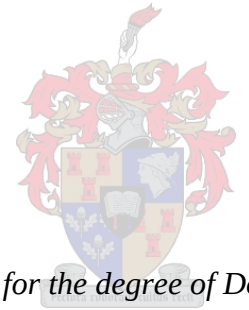
FEATURE SELECTION FOR MULTI-LABEL CLASSIFICATION

by Ivona E. Contardo-Berning



Dissertation presented for the degree of Doctor of Philosophy in the Faculty of Economic and Management Sciences at Stellenbosch University

Supervisor: Prof. S.J. Steel

December 2020

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights, and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Initials and Surname	Date
I.E. Contardo-Berning	December 2020

Abstract

The field of multi-label learning is a popular new research focus. In the multi-label setting, a data instance can be associated simultaneously with a set of labels instead of only a single label. This dissertation reviews the subject of multi-label classification, emphasising some of the notable developments in the field.

The nature of multi-label datasets typically means that these datasets are complex and dimensionality reduction might aid in the analysis of these datasets. The notion of feature selection is therefore introduced and discussed briefly in this dissertation. A new procedure for multi-label feature selection is proposed. This new procedure, relevance pattern feature selection (RPFS), utilises the methodology of the graphical technique of Multiple Correspondence Analysis (MCA) biplots to perform feature selection.

An empirical evaluation of the proposed technique is performed using a benchmark multi-label dataset and synthetic multi-label datasets. For the benchmark dataset it is shown that the proposed procedure achieves results similar to the full model, while using significantly fewer features. The empirical evaluation of the procedure on the synthetic datasets shows that the results achieved by the reduced sets of features are better than those achieved with a full set of features for the majority of the methods.

The proposed procedure is then compared to two established multi-label feature selection techniques using the synthetic datasets. The results again show that the proposed procedure is effective.

Keywords:

Multi-label Classification, Multiple Correspondence Analysis Biplots, Multi-label Feature Selection.

Opsomming

Die veld van multi-etiket leerteorie is ’n gewilde nuwe navorsingsarea. In die multi-etiket omgewing kan ’n datageval gelyktydig geassosieer word met ’n stel etikette in plaas van met slegs ’n enkele etiket. Hierdie verhandeling verskaf ’n oorsig oor die onderwerp van multi-etiket klassifikasie en beklemtoon sekere noemenswaardige ontwikkelings in die veld.

Die aard van multi-etiket datastelle leen homself tipies tot komplekse datasetelle waar dimensie reduksie die analise van hierdie datastelle kan vergemaklik. Die konsep van veranderlike seleksie word dus voorgestel en kortliks in hierdie verhandeling bespreek. ’n Nuwe prosedure vir multi-etiket veranderlike seleksie word voorgestel. Hierdie nuwe procedure, relevansie patroon verandelike seleksie (RPFS), maak gebruik van die metodologie van die grafiese tegniek van Meervoudige Ooreenstemmingsanalise bi-stippings om veranderlike seleksie uit te voer.

’n Empiriese evaluering van die voorgestelde tegniek is uitgevoer met behulp van ’n norm multi-etiket datastel en sintetiese multi-etiket datastelle. Vir die norm datastel word aangetoon dat die voorgestelde prosedure soortgelyke resultate lewer as die volledige model, maar met beduidend minder veranderlikes. Die empiriese evaluering van die prosedure op die sintetiese datastelle toon dat die resultate wat deur die gereduseerde stel veranderlikes gelewer word, beter is as dié wat met die volledige stel veranderlikes gelewer is, vir die meerderheid van die metodes.

Die voorgestelde prosedure word dan vergelyk met twee gevestigde multi-etiket verandelike seleksie tegnieke met behulp van die sintetiese datastelle. Die resultate toon weereens dat die voorgestelde prosedure effektief is.

Sleutelwoorde:

Multi-etiket Klassifikasie, Meervoudige Ooreenstemmingsanalise Bi-stippings, Multi-etiket Veranderlike Seleksie.

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance from many wonderful people. I would like to acknowledge the following people for their support and encouragement:

- My supervisor, Professor Sarel Steel, whose expertise and guidance were invaluable during each stage of the process.
- Professor Niel le Roux, particularly for his input in the formulation of the methodology. Thank you for being an inspiring lecturer and colleague.
- Dr Johané Nienkemper-Swanepoel for her valuable insights into the field of biplots and Correspondence Analysis.
- My husband, Tom, for his unfailing support and encouragement. I am extremely grateful.
- A special thank you to all our family, friends, and GraceLife family who offered support and encouragement along the way, especially to my dear friends Joep and Carol, Esther, and of course Rupert.

Finally, to my heavenly Father be the glory.

Table of Contents

List of figures

List of tables

List of appendices

Appendix A

Description of the 72 features of the *Emotions* dataset.

Appendix B

Comparison of relevance measures (correlation coefficient, Information Gain, and ReliefF) across four evaluation measures (Hamming-loss, One-error, Recall, and Precision) for all 24 datasets using SVM classifier.

Appendix C

Comparison of relevance measures (Correlation coefficient, Information Gain, and ReliefF) across four evaluation measures (Hamming-loss, One-error, Recall, and Precision) for all 24 datasets using XGBoost classifier.

Appendix D

Summary of results per dataset for SVM classifier.

Appendix E

Summary of results per dataset for SVM classifier.

Appendix F

Comparison per dataset of SVM and XGBoost classifier.

Appendix G

Three-way ANOVA per dataset.

Appendix H

Comparing the performance of techniques with respect to:

- Different signal-to-noise ratios
- The number of irrelevant features
- The label dependencies

- Different density vectors

Appendix I

Four-way ANOVA comparing:

- Dataset 1 vs Dataset 3
- Dataset 17 vs Dataset 21
- Dataset 1 vs Dataset 9
- Dataset 5 vs Dataset 17
- Dataset 1 vs Dataset 2
- Dataset 10 vs Dataset 12
- Dataset 17 vs Dataset 19
- Dataset 9 vs Dataset 17
- Dataset 16 vs Dataset 24

Appendix J

Comparison of feature selection techniques per dataset.

Appendix K

R code:

- Calculating the multi-label evaluation measures
- Implementation of RPFS procedure based on ReliefF using XGBoost classifier for the *Emotions* dataset
- Generating synthetic multi-label datasets
- Implementation of RPFS procedure based on IG using SVM classifier for synthetic datasets
- Implementation of FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013) on synthetic datasets

List of abbreviations

AD	Alzheimer's Disease
ADNI	Alzheimer's Disease Neuroimaging Initiative
AI	Artificial intelligence
ANOVA	Analysis of variance
ARFF	Attribute-Relation File Format
BPM	Beats per minute
BR	Binary relevance
BR+	Binary relevance Plus
CA	Correspondence Analysis
Dyn	Dynamic order
FS	Feature selection
FTT	Fourier Transform
IG	Information Gain
IQR	Interquartile range
kNN	k-Nearest Neighbours
Lasso	Least absolute shrinkage and selection operator
LP	Label powerset
LSI	Latent Sematic Indexing
MAP	Maximum a posteriori
MCA	Multiple Correspondence Analysis

MDDM	Multi-label Dimensionality Reduction via Dependence Maximisation
MFCC	Mel frequency cepstral coefficients
ML	Machine learning
ML-kNN	Multi-label k-Nearest Neighbours
MLSI	Multi-label Latent Semantic Indexing
MNO	Mobile Network Operators
MRI	Magnetic resonance imaging
NU	No update
OSH	Optimal separating hyperplane
PCA	Principal Component Analysis
PET	Positron Emission Tomography
PPT	Pruned Problem Transformation
PS	Probe selection
RAkEL	Random k-labelsets
RBF	Radial basis function
RPC	Ranking by pairwise comparison
RPFS	Relevance pattern feature selection
Stat	Static order
SVC	Support vector classifier
SVD	Singular Value Decomposition
SVM	Support vector machines

t-SNE	t-Distributed Stochastic Neighbour Embedding
XGBoost	Extreme gradient boosting

CHAPTER 1

INTRODUCTION

1.1 Background

The broad focus of this dissertation is statistical classification, an important component of supervised statistical learning. In the first chapter statistical learning is placed within the broader framework of Artificial Intelligence, Machine Learning, and Data Science. General aspects of statistical classification are discussed, and the focus is narrowed down to multi-label classification – one of the main themes of the dissertation. Section 1.3 is devoted to feature selection – the other main theme of the research. The chapter concludes with a statement of the problem addressed in the dissertation, and an overview of the dissertation is provided.

Recent advances in technology, and especially in computational power, have brought about a revolution in statistical analysis. One manifestation of this revolution is the onset of the *Big Data* era. Large amounts of data are continuously becoming available for analysis. Consequently, the scope for the use of statistical and machine learning techniques to solve real-life problems has increased dramatically. The research reported in this dissertation should be viewed from this perspective and it therefore seems appropriate to provide background on fields such as *Artificial Intelligence* (AI), *Machine Learning* (ML), and *Data Science*.

The field of AI originated in 1956 when John McCarthy invited leading researchers to the Dartmouth AI Conference in New Hampshire, U.S.A. (Solomonoff, 1985 and Moor, 2006). While AI is all the rage nowadays, it is interesting to note that current AI developments are vastly different from those previously imagined in science fiction books and movies. The *English Oxford Living Dictionary* defines AI as "The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages". At Amazon, AI is defined as "the field of computer science dedicated to solving cognitive problems commonly associated with human intelligence, such as learning, problem solving, and pattern recognition" (What is artificial intelligence?, n.d.). In essence, AI could be defined as human intelligence exhibited by machines.

ML is a widely used approach in AI. Marr (2016) defines ML as an application of AI based on the notion that machines could learn for themselves if one provides them with access to data. At its core, ML is a collection of algorithms that can be used to make predictions or classify data.

Although there are many definitions of Data Science on the internet (Data Science, n.d.) and in the literature (Cao, 2016), there seems to be consensus that it can broadly be defined as the overlap between Data Engineering (Computer Science), Mathematics and Statistics, as well as domain expertise or business knowledge. Data Science includes the entire data pipeline, *i.e.* the cleaning, processing, and transformation (wrangling) of data, visualisation of the data, as well as the application of appropriate algorithms. Each of these steps is vital to ensure the success of a Data Science project. Data Science enables companies to manipulate large amounts of data and to extract information in real-time in ways that were not anticipated a decade ago. It also allows for the extraction of value from new types of data such as social media data, web searches, and images.

The process of Data Science can be visualised as follows:

Figure 1.1 The process of Data Science.

It is important to note that this process is frequently iterative and often non-sequential, and that the skills and tools required to cover this process are varied.

Most of practical ML utilises supervised learning techniques. Supervised learning occurs when the data that have to be analysed contain input variables (the predictors or features) and one or more output variables (the response or target). In supervised learning an algorithm is used to learn the mapping function from the inputs to the output. The process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. The correct answers are known, and the algorithm iteratively makes predictions on the training data

and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

The following notation will be used in this dissertation. An input variable will be denoted by the symbol X. A vector containing $p > 1$ features will be denoted by $\mathbf{X}$, and the components of this vector will be denoted by $X_i, \quad i = 1, 2, \ldots, p$. Supervised learning allows one to make a prediction of the output Y, based on given values of the input variables contained in $\mathbf{X}$. Schematically an ML algorithm associates a predicted response with any given $\mathbf{X}$, *i.e.* $\mathbf{X} \rightarrow Y(\mathbf{X})$.

In broad terms a supervised learning procedure has two objectives. Firstly, a supervised procedure is frequently used for prediction purposes, as described in the previous paragraph. If this is the main objective, the available data are utilised in an algorithm to obtain a formula for predicting Y from $\mathbf{X}$. This typically entails determining the values of the parameters of the specific learning method by optimising an objective function. A second important objective of supervised learning is referred to as inference. The aim here is to obtain an indication of the importance of the different input variables when the response is predicted. It should be noted that these two objectives are frequently conflicting. Statistical models that are readily interpretable often do not give state-of-the-art results in terms of prediction accuracy. Conversely, many of the modern, highly accurate ML algorithms are of a black box nature, *i.e.* it is virtually impossible to obtain clear indications of the relative importance of the different input variables. This is in fact an important new area of research, *i.e.* finding ways to interpret the output from an accurate black box predictor. Refer to Baehrens *et al*. (2010), Lipton (2017), and Weng (2017) for more information on this topic.

Supervised learning problems can be grouped into regression and classification problems depending on the nature of the response variable.

- **Classification problems**: The output variable is qualitative or categorical, with categories such as "*retain*" and "*churn*", or "*disease*" and "*no disease*".

- **Regression problems**: The output variable is quantitative or numerical, assuming real values, such as "*profit*" or "*house price*".

In the case of a quantitative output, one aims to find a function $f(\mathbf{X})$ that can be used to predict Y given the values of the input $\mathbf{X}$. This requires a loss function $L(Y, f(\mathbf{X}))$, which penalises errors in prediction. If the predictions are inaccurate, the loss function will output a higher number than when the predictions are accurate. The most popular loss function is simply the *square error loss*: $L(Y, f(\mathbf{X})) = (Y - f(\mathbf{X}))^2$. When performing supervised learning, one wishes to minimise the expected value of the loss function.

Two considerations when performing supervised learning are the complexity of the model fitted to the data, and the trade-off between the bias and the variance of the model. These two are related. A high level of model complexity will lead to a model which overfits the data. Overfitting means that the learner fits the training dataset very well but does not perform well when asked to predict new unseen instances. The bias-variance trade-off also relates to the ability of the model to generalise to new instances. A decrease in the bias will lead to higher variance, and *vice versa*. A model with high bias and low variance would be a model that is consistently wrong 25 per cent of the time. A model with low bias and high variance can be wrong anywhere between 5 and 40 per cent of the time, depending on the dataset that is used to train the model.

Examples of popular algorithms used in supervised learning are support vector machines, neural networks, logistic regression, random forests, and decision trees.

Turning to unsupervised learning, one only has observations available on several input variables, *i.e.* there is no response variable or "teacher" to guide the learning process. There are several goals in unsupervised learning. One of the main objectives is to estimate the underlying multivariate distribution that generated the data. This is usually too ambitious. Instead, regions in the input space having high probability are often identified. Another frequent objective in unsupervised learning is that of dimension reduction. Examples of unsupervised techniques used for this purpose are principal components, principal curves and self-organising maps.

Three of the main problems addressed in unsupervised learning are as follows:

- **Clustering problems**: The inherent groupings in the data are important, such as grouping clients of a bank by purchasing behaviour or risk.

- **Anomaly detection problems**: Unsupervised learning can be used to identify outliers in a dataset, for example banks will be able to detect fraudulent transactions by identifying unusual patterns in the credit card use of a customer.

- **Association problems**: Here it is of interest to discover rules that describe large portions of the data, for example customers who buy Product A also tend to buy Service B.

This research is concerned with supervised learning problems and specifically classification problems. Due to its central role in this research, classification will be discussed in more detail in the next section.

1.2 Classification

Consider a training dataset $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, N\}$ consisting of N observations on p input variables, $X_1, X_2, \ldots, X_p$, and corresponding observations on a categorical response variable Y. In classification the main objective is to use the training data to construct a *classifier*. A classifier is a formula or rule that can be used to assign an item (instance) to one of the available groups based on the values of $X_1, X_2, \ldots, X_p$ measured for the item. If $\mathbf{x}$ denotes these measurements, a classifier will therefore be a function of $\mathbf{x}$. A good classifier is one which accurately predicts such class memberships.

It is possible to construct a hierarchy of classification problems based on the number of response variables and their nature. Let q denote the number of response variables (also referred to as labels) and K the number of categories per response variable. If there are more than one response variable, it is assumed that they all have the same number of categories. Four categories of classification problems can now be distinguished, *viz.* binary, multi-class, multi-label, and multi-class multi-label (or multi-output) problems. This hierarchy of classification is summarised in Table 1.1. Each of these scenarios is discussed in more detail below Table 1.1.

Table 1.1: Classification problems.

	Number of labels per instance q		
Number of classes in each label (K)		$q = 1$	$q \geq 2$
	$K = 2$	Binary	Multi-label
	$K \geq 3$	Multi-class	Multi-class multi-label

Single-label classification problems, where $q = 1$, are characterised by the presence of only one response variable. If in addition $K = 2$, the resulting problem is one of binary classification – the classification of items into one of two groups. Figure 1.2 provides a graphical representation of a binary classification problem, assuming that there are only two input variables. As an example, the data could represent patients who have to be classified into one of two groups, where blue in Figure 1.2 denotes patients who suffer from the disease and red those who do not. Also shown in Figure 1.2 is a classification function that is linear in the two input variables.

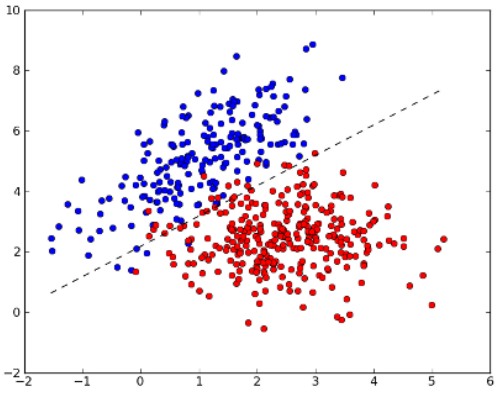


Figure 1.2 Binary classification with a linear classifier.

If $q = 1$ and $K > 2$, the resulting problem is referred to as a multi-class classification problem. For multi-class classification the interest lies in classifying items into one of several (more than two) groups. As an example, new clients may have to be classified into one of five different credit score categories based on their payment history, disposable income, and credit usage. Table 1.2 provides an illustration.

Table 1.2: Multi-class classification.

Client No.	Credit Risk Category				
	Very Poor	**Poor**	**Fair**	**Good**	**Excellent**
1	x				
2				x	
3		x			

It is important to note that each client is classified into a single category only, *i.e.* $q = 1$.

Multi-label problems are characterised by the presence of several response variables (labels), denoted by $Y_1, Y_2, \ldots, Y_q$ where . $q \geq 2$ .. These labels are summarised in a q -component vector $\mathbf{Y}$. In such problems the training data are therefore given by $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \ldots, N\}$. The simplest instance of such a multi-label scenario is where each $Y_j \in \{0,1\}$, $j = 1, 2, \ldots, q$. In this case $Y_j = 1$ implies the presence of label j, while $Y_j = 0$ implies its absence. An example of such a scenario is musical instrument recognition. In this case, each data item is a music piece. The response variables represent the different musical instruments that may be playing the piece, and the multi-label classification task entails training a model that can (accurately) identify the instruments playing a given piece. The input vectors will typically contain the features that can be used to differentiate between instruments, for example Mel frequency cepstral coefficients, and different measures of rhythm, loudness, and timbre.

Multi-label classification problems where each label is binary will be the focus of the research reported in this dissertation.

The difference between single-label and multi-label classification problems is illustrated in Tables 1.3 and 1.4 below. The structure of a single-label problem with four input variables is shown in Table 1.3.

Table 1.3: Single-label classification problem.

X_1	X_2	X_3	X_4	Y
-1.7	2.8	4.1	0.1	1
-2.2.	-3.7	0.4	-3.6	0
3.4	0.9	-0.3	-1.0	1

Table 1.4 illustrates the structure of a multi-label problem with four input variables and four labels.

Table 1.4: Multi-label classification problem.

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
-1.7	2.8	4.1	0.1	1	0	0	1
-2.2.	-3.7	0.4	-3.6	0	1	1	0
3.4	0.9	-0.3	-1.0	0	1	1	1

It is clear that multi-label classification problems have a much richer content than single-label problems.

The distinguishing characteristic of a multi-label classification problem is that a single instance can be assigned multiple labels. Consequently, multi-label classification has traditionally been applied mostly in the categorisation of text data or in medical diagnosis. A text document usually belongs to more than one category. For example, an article relating to the effect of the recent drought in the Western Cape on the South African growth outlook can be assigned each of the labels *South African economy*, *drought*, and *Cape Town*. Similarly, in medical diagnosis a patient may be suffering from heart disease and diabetes simultaneously.

More recent applications of multi-label classification are protein function classification (Luo and Zincir-Heywood, 2005 and Barros *et al*., 2013), music categorisation (Li and Ogihara, 2003), semantic scene classification (Boutell *et al.*, 2004), video annotation (Qi *et al.*, 2007), image classification (Qi *et al.*, 2009), food truck recommendation (Rivolli *et al*., 2017, and Rivolli *et al.*, 2018), and text data classification (see for example the Magpie Text Classification project (Stypka, 2015) on GitHub). These and other applications have led to an increase in interest in the field of multi-label classification.

The last category in the classification hierarchy is multi-class multi-label classification. Here the aim is to assign one or more labels to each instance, but now each label is associated with one of $K > 2$ possible classes. Table 1.5 illustrates the structure of such a classification problem.

Table 1.5: Multi-class multi-label classification problem.

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
-1.7	2.8	4.1	0.1	1	0	0	2
-2.2.	-3.7	0.4	-3.6	0	2	3	0
3.4	0.9	-0.3	-1.0	0	4	3	1

Consider the following applications of multi-label multi-class classification: In the categorisation of documents each document in a corpus (for example, an article available on Wikipedia) is associated with one or more topics (for example, the categories at the bottom of every Wikipedia article). In this particular scenario, the set of classes is extremely large. For example, if one visits the Wikipedia article on "*Brexit*", there are multiple labels present, such as *Brexit*, *2010s in politics*, *2016 in British politics*, *etc*. Some of these labels contain more than one class, such as *2010s in politics*, which contains classes *2011*, *2012*, *etc*. The field of computer vision allows for the classification of images into semantic classes such as *mountains*, *trees* or *dogs*. The label *dogs* could contain multiple classes such as *English Cocker Spaniel*, *Great Dane* or *German Shepherd*.

The categorisation of documents (Dekel and Shamir, 2010), computer vision or semantic scene classification (Boutell *et al.*, 2004), and gene function prediction in computational biology (Barutcuoglu *et al.*, 2006) are all examples of multi-class multi-label classification problems. Multi-class multi-label problems are the most complex of the classification problems.

In the following section a brief background on feature selection will be provided, including some comments on the application of feature selection within a multi-label context.

1.3 Feature selection

Guyon and Elisseeff (2003) argue that the term *variable* refers to a raw input variable, while the term *feature* refers to a variable constructed from the raw input variables. In this dissertation the term feature will, from now on, be used throughout.

Consider a supervised learning scenario with input features $\{X_1, X_2, ..., X_p\}$. Feature selection (FS) aims to find a small number of features that provide as much information about the dataset as the original set of features did. FS is able to effectively reduce the dimension of the data by

removing irrelevant and/or redundant features. *Irrelevant* features are features that do not provide any useful information to the classifier, whereas *redundant* features do contain useful information, but this information is already contained in one or more of the other features in the dataset.

During FS, the feature space $X = \{X_1, X_2, ..., X_p\}$ is searched to find a subset $X' \subseteq X$ such that X' describes the dataset almost as well as X does. This will lead to faster learning algorithms and sometimes even improved performance of learning algorithms. A number of studies show that features can frequently be omitted without a large sacrifice in performance.

Using the subset of features obtained from FS allows for improved interpretation of the final model. Identifying the most important feature is of particular importance in medical applications, where there are potentially large costs associated with the measurement of features.

FS has been studied extensively in the single-label context, but few results in FS on multi-label learning have been reported (Spolaôr *et al.*, 2013). The single-label learning algorithms do not perform well when applied directly in the multi-label setting. In this dissertation, a new procedure for performing FS in the multi-label context is proposed.

In Section 1.4, the problem statement will be presented and a brief description of the proposed procedure to solve it will be provided.

1.4 Problem statement

First, consider the problem of FS in a multi-label setting. FS aims to identify the features that are relevant or important when labels are assigned to a new data case. This is a more complicated problem than in single-label scenarios, since a feature may be relevant when considering assigning, say, label 1, while it is largely irrelevant when a decision has to be made regarding the assignment of the other labels.

Consider the input features $X_1, X_2, ..., X_p$ and labels $1, 2, ..., q$. Which of these p features are relevant? How does one determine relevance? How does one approach this problem? In order to answer these questions, one needs to consider the difference between the *local* and the *global* relevance of a feature as argued by Sandrock and Steel (2016).

A first naïve approach would be to declare a feature relevant for the scenario if it is relevant for at least one of the labels. Sandrock and Steel (2016) define a feature to be locally relevant for a given label if it explains the label, irrespective of its relevance for any of the other labels. This discussion is made easier if some notation is introduced.

Let $\mathbf{A}: p \times q = \left[A_{ij} \right]$ be a matrix with entries

$$A_{ij} = \begin{cases} 0 & \text{if feature } i \text{ is deemed irrelevant for label } j \\ 1 & \text{if feature } i \text{ is deemed relevant for label } j \end{cases}.$$

The entries in such a matrix will depend on the method that is used to decide whether X_i is relevant for label j. In this study, three measures of the strength of the relationship between the feature and the label are investigated to determine relevance. These are the correlation coefficient, ReliefF, and Information Gain. These relevance measures will be discussed in Section 3.2.2. The row totals $A_{i+} = \sum_{j=1}^{q} A_{ij}$, $i = 1, 2, ..., p$ provide useful information regarding the overall relevance of the features, and the naïve approach mentioned above would declare feature i globally relevant if $A_{i+} > 0$. A feature is globally relevant if it is deemed relevant for several or all of the labels.

An interesting possibility is to consider the selection as a two-step procedure. In the first stage, a grouping of the features into non-overlapping groups is performed, with the features in a given group similar to one another. During the second stage of this process FS is performed separately for each of the feature groups. How does one go about grouping the features? This entails a grouping of the features and one requires a measure quantifying the similarity between features.

This dissertation will focus on introducing a novel approach to FS in the multi-label context. The approach applies well-known techniques to a multi-label problem. The method that will be proposed uses the established method of Multiple Correspondence Analysis (MCA) – an extension of Correspondence Analysis (CA) – to perform the initial grouping of the features based on the relevance matrix obtained using the three relevance measures (the correlation coefficient, Information Gain, and ReliefF). The MCA biplot enables one to group features together that provide similar information (*i.e.* that lie close together on the biplot). During the second stage, one can then utilise the inherent ranking abilities of the relevance measures to

identify the features that rank highest in each of the feature groups. A more detailed description of the proposal is provided in Chapter 3.

The recent increased interest in the field of multi-label classification and the diverse practical applications implies that the development of a new FS technique could have significant practical implications. Firstly, CA, MCA, and biplots are established and powerful statistical techniques. Biplots are used extensively to provide graphical representations of complex multivariate datasets, but to date biplots have not been used for FS.

The use of the MCA biplot methodology to perform FS provides the practitioner with a visual representation of the associations between the features and the labels. The technique is able to distinguish between irrelevant and redundant features. Features that provide similar information are grouped together, and features can be ranked according to importance within these groupings.

1.5 Outline

This dissertation is divided into six chapters, where the first chapter provides a brief introduction to the field of multi-label classification and multi-label feature selection, as well as a clear description of the problem statement and the contribution of this research. Chapters 2 and 3 provide a theoretical framework for this study, and Chapters 4 and 5 describe the empirical work done in this dissertation.

Chapter 2 discusses general aspects of multi-label classification and the evaluation measures that are used to evaluate multi-label classification techniques. Some of the most popular multi-label classification approaches are presented and described. Finally, the base classifiers used in this dissertation, namely support vector machines (SVMs) and extreme gradient boosting (XGBoost) are summarised.

The focus in Chapter 3 is on multi-label FS. In this chapter, the approaches to FS in the multi-label context are discussed with specific attention to the methods proposed by Sandrock and Steel (2016) and Spolaôr *et al.* (2013). Finally, a new FS method, namely, relevance pattern feature selection (RPFS) is proposed that utilises aspects of MCA biplot methodology to address the problem of multi-label FS.

The results pertaining to the empirical investigation on one of the benchmark datasets are presented in Chapter 4. In this analysis, FS is performed on the *Emotions* dataset using RPFS. Classification using the two multi-label classifiers, namely SVM and XGBoost, is performed on the resulting reduced datasets. The results for the two base classifiers are first discussed individually and then compared at the end of the chapter.

In Chapter 5, the empirical investigation is repeated on 24 synthetic multi-label datasets. The chapter starts by discussing the methods available for generating synthetic multi-label datasets, the properties of these datasets as well as the cases that are considered in this dissertation. Secondly, the FS technique is applied to each of the 24 synthetic multi-label datasets, and the SVM classifier is used to perform the analysis. This is repeated using the XGBoost classifier. The results are again compared. In the final section, RPFS is compared with the FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al.* (2013).

The final chapter presents a summary of the conclusions that can be drawn from the empirical investigation and some recommendations for future research are made.

CHAPTER 2

MULTI-LABEL CLASSIFICATION

2.1 Introduction

With the current surge in popularity of the fields of AI, ML, and Data Science, the interest in classification algorithms has increased. In the context of multi-label classification, there has been a marked increase in the amount of research over the past decade. In music classification, a song can be categorised in terms of more than one emotion (Li and Ogihara, 2003, Trohidis *et al*. 2008, and Trohidis *et al*., 2011), or it can belong to more than one genre, for example *Viva la Vida* by the British Rock band *Coldplay* can be labeled as *British Pop, Alternative Rock,* and *Indie Rock*. Multi-label classification applied to the Alzheimer's Disease Neuroimaging Initiative (ADNI) database shows that new multi-label research leads to improved performances of both clinical score prediction and disease stage identification (Cheng *et al.*, 2015). A photograph can belong to more than one category at the same time, for example *sunrise, boat*, and *ocean* (Boutell *et al.*, 2004).

As mentioned in Section 1.2, due to the importance of the role of classification in this dissertation, the topic will be discussed in more detail in this chapter. Firstly, a discussion of the important concepts pertaining to multi-label classification will be presented. This will include a description of the aim of multi-label classification, the characteristics of multi-label datasets, and the multi-label benchmark datasets that are available for analysis. The multi-label evaluation measures that allow for the evaluation and comparison of the performance of multi-label classification techniques are discussed based on an example in Section 2.3. In Sections 2.4 to 2.6, the relevant research in the field of multi-label classification is presented with consideration of the problem transformation, algorithm adaptation, and ensemble approaches. A discussion of the two machine learning algorithms used in this study will be provided in Section 2.7. Finally, a summary of the literature review on multi-label classification will be given in Section 2.8.

2.2 General aspects of multi-label classification

In this section, the task of multi-label learning is defined, and definitions of all key concepts required to understand the problem are provided, while paying specific attention to the aspect of dependence amongst the labels. Finally, the multi-label benchmark datasets are introduced.

2.2.1 Aim of multi-label classification

Multi-label learning problems are concerned with learning from instances where each instance is simultaneously associated with multiple labels. Depending on the goal of classification, one can distinguish between two types of classification problems: multi-label *classification* problems and multi-label *ranking* problems (Tsoumakas *et al.*, 2010). In multi-label classification the aim is to construct a model that will predict a list of relevant labels for a new, unseen instance. The aim of multi-label ranking is to construct a model that will provide a list of preferences of labels from the set of possible labels for a given, new, unseen instance (Madjarov *et al.*, 2012). A discussion of the characteristics of multi-label datasets is required prior to presenting proposed solutions to these two problems. In the next section, the concepts of label cardinality, label density, and label dependence will be defined.

2.2.2 Characteristics of multi-label datasets

In some real-world multi-label datasets, there are a large number of labels associated with each example, and in others there are only a small number of labels for each example. This aspect or characteristic of the datasets can influence the performance of different multi-label techniques (Tsoumakas *et al.*, 2010). Based on this, the two main characteristics of a multi-label dataset are the label cardinality and label density. Both measures relay information regarding the number of labels of a multi-label dataset.

Label density and label cardinality

In order to compare multi-label datasets, the concepts of label density and label cardinality need to be introduced. Consider the notation presented in Chapter 1: a training dataset $\{(\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \ldots, N\}$ consisting of N observations on p input features, $X_1, X_2, \ldots, X_p$ and corresponding observations on several categorical response variables.

The label cardinality refers to the average number of labels of the instances in a multi-label dataset:

$$\text{Label cardinality} = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{Y}_i|,$$

where $|\mathbf{Y}_i|$ is the sum of the components of the vector. While the label cardinality does not explicitly take the number of labels into account, the label density does so. The label density is defined as the average number of labels of the examples, divided by the number of labels, q:

$$\text{Label density} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\mathbf{Y}_i|}{q}.$$

Tsoumakas *et al*. (2010) argue that two datasets with the same label cardinality but with very different label densities might not present the same properties. The performance of multi-label learning methods could also be influenced by these characteristics. For example, many of the transformation algorithms discussed in Section 2.4 operate on a subset of labels which are directly influenced by the number of distinct labels. Bernadini *et al*. (2014) investigate whether these parameters influence the performance of different multi-label methods using artificial datasets. Refer to Bernadini *et al*. (2014) for more details.

A discussion of the characteristics of multi-label problems is not complete without mentioning the aspect of label dependence.

Label dependence

The inherent label dependence which is associated with multi-label datasets provides for a richer and more complex scenario than the simple single-label classification problems considered in the past. The categories or labels in many real-world applications are not independent. For example, in semantic scene classification, one would expect labels such as *boat*, *sail*, and *ocean* to be dependent and to occur simultaneously. When there is a strong dependency among labels, there might well be an advantage in leveraging these dependencies (Ghamrawi and McCallum, 2005).

One could argue that label dependence should be included at all cost, but it is important that both the computational cost and complexity of the resulting classifier are considered. This becomes even more important in high-dimensional problems. One fundamental question that one should consider is: Should a classifier take label dependencies into account when estimating $P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x})$? This leads to a second question: If these conditional label dependencies can be estimated accurately, will it lead to a better classification, or will it simply add unnecessary complexity? The answers to these questions lie in whether or not these dependencies can be estimated accurately. It seems likely that an inaccurate estimation of the label dependence would lead to a less accurate classifier.

Read (2013) notes that determining whether label dependencies are present can prove to be problematic. The author introduces the following definitions to aid the discussion. If the joint distribution of labels is not a product of the marginal distributions, then a so-called unconditional dependence exists, namely

$$P(Y_j, Y_k) \neq P(Y_j) \times P(Y_k) .$$

Conditional dependence of labels given the input features, $\mathbf{x}$, exists when

$$P(Y_j, Y_k \mid \mathbf{X} = \mathbf{x}) \neq P(Y_j \mid \mathbf{x}) \times P(Y_k \mid \mathbf{x}) .$$

One can quantify the extent of the unconditional label dependence by using, for example, the observed label frequencies to estimate the mutual information. For more details on this, see Read (2013). Measuring conditional label dependence is significantly more challenging. How can one accurately estimate the label dependence structure when there are a large number of predictors? Estimating the conditional label dependence is a difficult task, which will not be included in this dissertation. For more information on the topic of label dependence in multi-label classification, refer to Dembczyński *et al*. (2010) and Dembczyński *et al*. (2012).

2.2.3 Benchmark multi-label datasets

The majority of the results from multi-label research is tested on a number of benchmark datasets, which can be found on the MULAN website (Tsoumakas *et al.*, 2011). The 26 datasets available on MULAN are summarised in Table 2.1.

Table 2.1: Benchmark datasets from MULAN.

Name	Domain	Instances	Nominal features	Numeric features	Labels	Cardinality	Density
Bibtex	text	7395	1836	0	159	2.402	0.015
birds	audio	645	2	258	19	1.014	0.053
bookmarks	text	87856	2150	0	208	2.028	0.010
CAL500	music	502	0	68	174	26.044	0.150
corel5k	images	5000	499	0	374	3.522	0.009
corel16k	images	13811	500	0	161	2.867	0.018
Delicious	text (web)	16105	500	0	983	19.020	0.019
Emotions	music	593	0	72	6	1.869	0.311
Enron	text	1702	1001	0	53	3.378	0.064
EUR-Lex (1)	text	19348	0	5000	412	1.292	0.003
EUR-Lex (2)	text	19348	0	5000	201	2.213	0.011
EUR-Lex (3)	text	19348	0	5000	3993	5.310	0.001
Flags	images (toy)	194	9	10	7	3.392	0.485
Genbase	biology	662	1186	0	27	1.252	0.046
Mediamill	video	43907	0	120	101	4.376	0.043
Medical	text	978	1449	0	45	1.245	0.028
NUS-WIDE	images	269648	0	128	81	1.869	0.023
rcv1v2 (1)	text	6000	0	47236	101	2.880	0.029
rcv1v2 (2)	text	6000	0	47236	101	2.634	0.026
rcv1v2 (3)	text	6000	0	47236	101	2.614	0.026
rcv1v2 (4)	text	6000	0	47229	101	2.484	0.025
rcv1v2 (5)	text	6000	0	47235	101	2.642	0.026
Scene	image	2407	0	294	6	1.074	0.179
tmc2007	text	28596	49060	0	22	2.158	0.098
yahoo	text	5423	0	32786	31	1.481	0.051
yeast	biology	2417	0	103	14	4.237	0.303

Three of the most popular benchmark datasets are the *Emotions*, *Scene*, and *Yeast* datasets. The *Emotions* dataset consists of 593 songs which are labelled according to six emotions: *Happy-Pleased*, *Amazed-Surprised*, *Relaxing-Calm*, *Quiet-Still*, *Sad-Lonely*, and *Angry-Aggressive* (Li and Ogihara, 2003). The *Emotions* dataset will be discussed in greater detail in Section 4.3. The *Scene* dataset consists of 2407 images annotated with six labels, namely *Beach*, *Sunset*, *Fall foliage*, *Field*, *Mountain*, and *Urban* (Boutell *et al*., 2004). Biological functions form the 14 labels of the *Yeast* dataset. The 2417 genes can be associated with these 14 functional classes (Elisseeff and Weston, 2001). Examples of these functional classes are *Metabolism*, *Protein synthesis*, *Cellular biogenesis*, and *Transcription*.

On MULAN, these benchmark datasets are all in ARFF (Attribute-Relation File Format) and have an associated XML file which describes the labels. The datasets cover several different

domains, for example text, images, and music. They also cover different levels of dimensionality, which can be particularly useful for evaluating multi-label feature selection techniques. However, they are limited with regard to label cardinality and label density.

Other repositories that host multi-label datasets are KEEL (Alcalá-Fdez *et al*. (2009) and Alcalá-Fdez *et al*. (2011)), MEKA (Read *et al*., 2016), and RUMDR (Charte *et al*., 2018).

In the next section, the evaluation measures that quantify the performance of multi-label classification procedures will be presented.

2.3 Evaluation measures

The evaluation of multi-label classifiers is a more complex problem than in the single-label case. In the single-label scenario, one is typically only interested in which of the predicted cases were classified correctly, and which were not. In the multi-label setting, one would like to evaluate the labels dependently, and for this reason, the goal is to measure how well the classifier predicts a joint set of labels, instead of evaluating each label separately.

Bipartitions-based evaluation measures are calculated by comparing the labels that are *predicted* to be relevant to the labels that are *known* to be relevant. There are two sets of methods available to evaluate the bipartitions. The first group of methods – example-based methods – is calculated based on the average differences between the actual and predicted sets of labels over all instances in the test dataset. The second group of methods – label-based methods – decomposes the evaluation process into separate evaluations for each individual label. One could also use ranking-based measures which compare the predicted ranking of the labels to the true rankings. This dissertation will only consider the example-based and ranking-based methods. For more information on the label-based methods, refer to Tsoumakas *et al*. (2010).

Figure 2.1 provides a visual representation of the evaluation measures that are available.

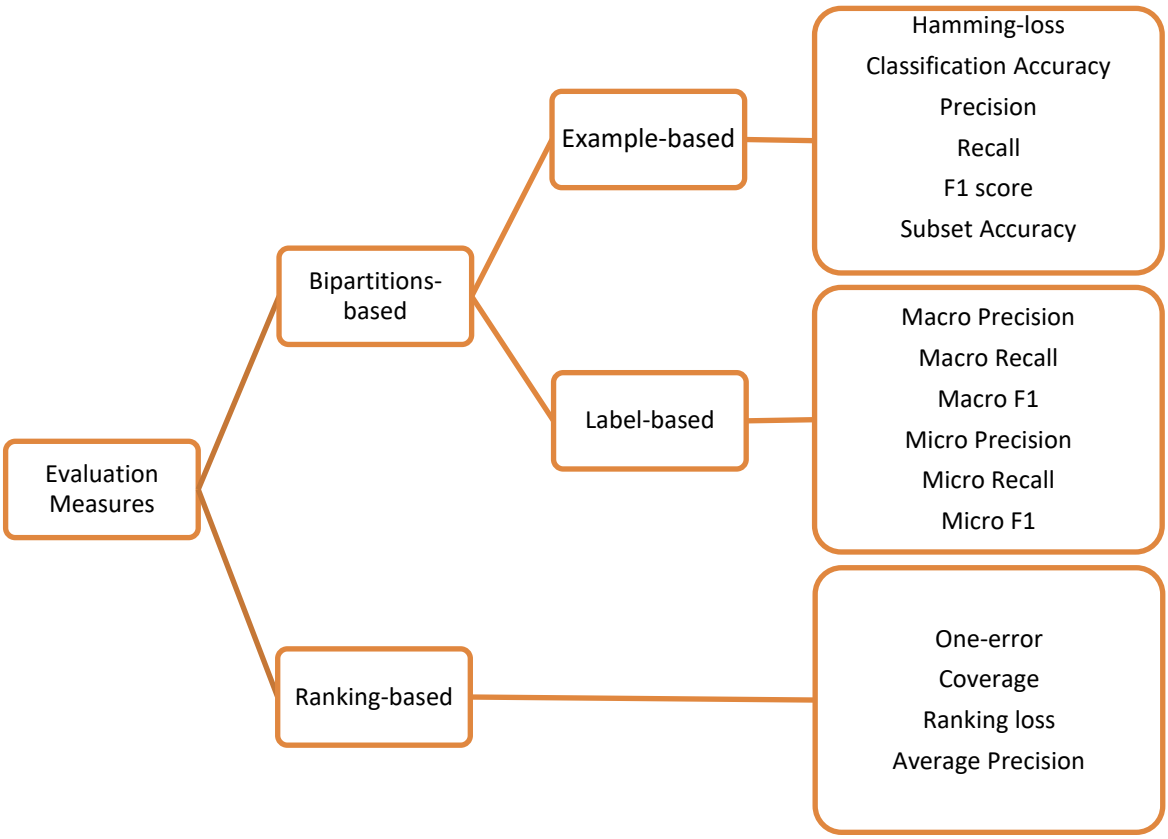


Figure 2.1 Evaluation measures for multi-label classification.

2.3.1 Example-based evaluation measures

In order to aid the discussion of the example-based methods, consider a multi-label dataset with N observations, and six labels, $q = 6$. The predicted labels and rankings of two classifiers on the same instance are presented in Table 2.2. Note that the predicted labels are denoted by the binary variables $Z_1, Z_2, \ldots, Z_q$. Obtaining these predictions will depend on the specific multi-label approach that is used – this will be explained later.

Table 2.2: Evaluating the performance of two classifiers.

Classifier	True labels $\mathbf{Y}_i$	Predicted labels $\mathbf{Z}_i$	Predicted ranks $\mathbf{r}_i$
1	$[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$	$[1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1]$	$[2 \quad 1 \quad 6 \quad 4 \quad 5 \quad 3]$
2	$[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$	$[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$	$[1 \quad 2 \quad 3 \quad 4 \quad 6 \quad 5]$

The true labels for case i are $\mathbf{Y}_i$ and the predicted (assigned) labels for case i, $\mathbf{Z}_i$, with the rank assigned to label k denoted by $\mathbf{r}_i(k)$, $k = 1,...,q$. The most relevant label according to the classification receives a rank of one, while the least relevant label receives a rank of q.

The *Hamming-loss* evaluation measure (Schapire and Singer, 2000) is the average proportion of labels that are misclassified and is defined as follows:

$$\text{Hamming-loss} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{q}\frac{|\, y_{ik} - z_{ik} \,|}{q}.$$

This measure provides an indication of how many times a label not belonging to the subset of correct labels for the instance is predicted to be present, or how many times a label belonging to the subset of correct labels is not predicted to be present. Since Hamming-loss evaluates each label separately, it resembles the misclassification error in the binary classification context. Lower values of Hamming-loss indicate better performance, where perfect performance (based on this measure) results in a Hamming-loss of zero.

Classification Accuracy (Zhu *et al.*, 2005) or subset accuracy (Ghamrawi and McCallum, 2005) is defined as

$$\text{Classification Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I\left(\mathbf{Z}_i = \mathbf{Y}_i\right),$$

where $I(\text{true}) = 1$ and $I(\text{false}) = 0$. This is a very strict evaluation measure as it requires the predicted label set to be an exact match to the true label set. For a dataset with a large number of labels, this measure will not provide valuable information about the performance of the classifier, since in such cases even highly accurate classifiers may be incorrect for a few of the labels.

Godbole and Sarawagi (2004) suggest the use of Precision, Recall, F_1, and Accuracy as evaluation measures. *Precision* estimates the expected proportion of predicted labels that are correct (the average proportion of predicted labels that are correct) as follows:

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\, \mathbf{Y}_i \cap \mathbf{Z}_i \,|}{|\, \mathbf{Z}_i \,|}.$$

Recall estimates the expected proportion of true labels that are correctly predicted (the average proportion of true labels that are correctly predicted), *i.e.*

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i|}.$$

Precision and Recall are similar to the concepts of *Sensitivity* and *Specificity* used especially in medical classification problems. These are defined as follows:

- Sensitivity refers to the probability of predicting disease given the true state is disease. This corresponds to Recall.
- Specificity refers to the probability of predicting non-disease given the true state is non-disease. Specificity (the true negative rate) is related to Precision (the positive predictive value). The exact relationship depends on the percentage of positive cases in the population.

Since a trade-off exists between these two measures – an increase in one of these measures usually occurs at the expense of a decrease in the other – they should ideally be considered simultaneously. To this end the measure . F_1 . is the harmonic mean of the two, *i.e.*

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}.$$

Accuracy is calculated as

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i \cup \mathbf{Z}_i|}$$

Accuracy is determined by expressing the labels that are predicted correctly as a proportion of the labels in $\mathbf{Y}_i$ or $\mathbf{Z}_i$. Accuracy considers both the actual and the predicted label sets simultaneously.

To illustrate these measures, consider the vectors of actual and predicted labels given in Table 2.2. Smaller values of Hamming-loss mean better performance, whereas higher values of Classification Accuracy, Precision, Recall, F_1-score, and Accuracy correspond to better

performance. The evaluation measure which is highlighted in bold in Table 2.3 corresponds to the best classifier for the evaluation measure.

Table 2.3: Evaluating the performance of two classifiers using example-based measures.

Classifier	**1**	**2**
True labels $\mathbf{Y}_i$	$[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$	$[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$
Predicted labels $\mathbf{Z}_i$	$[1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1]$	$[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$
Hamming-loss	**0.33**	0.5
Classification Accuracy	**0**	**0**
Precision	**0.67**	0.50
Recall	0.67	**1**
$\boldsymbol{F_1}$ **score**	**0.67**	**0.67**
Accuracy	**0.50**	**0.50**

From Table 2.3, one could argue that Classifier 1 performs better than Classifier 2, due to the fact that it performs better than Classifier 2 based on Hamming-loss and Precision, while Classifier 2 only performs better based on Recall. Usually, there is much more room for interpretation when comparing different classifiers. Typically, it will depend on the specific problem. For example, if it is important to consider both the actual and the predicted label sets simultaneously, one could give a greater weight to Accuracy when comparing different classifiers.

In Section 2.3.2 the ranking-based evaluation measures are described using the same example as above.

2.3.2 Ranking-based evaluation measures

The ranking-based measures evaluate the accuracy of the label ranking provided by the multi-label classifier. *One-error* determines whether the top-ranked label is relevant or not and ignores the relevancy of all other labels. Here relevancy implies that the label is in $\mathbf{Y}_i$. One-error can take on values between zero and one, and a smaller value indicates better performance. It is defined as

$$\text{One-error} = \frac{1}{N}\sum_{i=1}^{N}\delta\left(\arg\min_{k\in Z_i} r_i(k)\right)$$

where

$$\delta(k) = \begin{cases} 1 & \text{if } k \notin \mathbf{Y}_i \\ 0 & \text{otherwise} \end{cases}.$$

Coverage evaluates how far, on average, one needs to go down the ranked list of labels in order to cover all the relevant labels of the example. Coverage is defined as

$$\text{Coverage} = \frac{1}{N}\sum_{i=1}^{N}\frac{\max_{k\in\mathbf{Y}_i} r_i(k)}{|\mathbf{Y}_i|} - 1.$$

Smaller values of Coverage imply better performance.

The *Ranking-Loss* measures the average fraction of pairs of labels which are not correctly ordered, *i.e.*:

$$\text{Ranking-Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathbf{Y}_i||\bar{\mathbf{Y}}_i|}\left|\left\{(k_a,k_b): r_i(k_a) > r_i(k_b), (k_a,k_b)\in\mathbf{Y}_i\times\bar{\mathbf{Y}}_i\right\}\right|,$$

where $\bar{\mathbf{Y}}_i$ is the complementary set of $\mathbf{Y}_i$ with respect to L. Note that $L = \{1,2,...,q\}$ is the set of all labels.

Average Precision calculates for each relevant label, the percentage of relevant labels among all labels that are ranked above it, and then averages over all relevant labels. It evaluates the average fraction of labels ranked above a particular label $k \in \mathbf{Y}_i$ which actually are in $\mathbf{Y}_i$. This quantity is therefore given by

$$\text{Average Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathbf{Y}_i|}\sum_{k\in\mathbf{Y}_i}\frac{\left|\left\{k'\in\mathbf{Y}_i : r_i(k') \le r_i(k)\right\}\right|}{r_i(k)}.$$

Ideally, Average Precision should be as large as possible.

Smaller values of One-error, Coverage, and Ranking-loss mean better performance, whereas higher values of Average Precision correspond to better performance. The evaluation measure

which is highlighted in bold in Table 2.4 corresponds to the best classifier for the evaluation measure.

Table 2.4: Evaluating the performance of two classifiers using ranking-based measures.

Classifier	**1**	**2**
True labels $\mathbf{Y}_i$	$[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$	$[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$
Predicted labels $\mathbf{Z}_i$	$[1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1]$	$[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$
Predicted ranks $\mathbf{r}_i$	$[2 \quad 1 \quad 6 \quad 4 \quad 5 \quad 3]$	$[1 \quad 2 \quad 3 \quad 4 \quad 6 \quad 5]$
One-error	**0**	1
Coverage	1	**0.67**
Ranking-loss	**0.44**	**0.44**
Average Precision	**0.72**	0.59

From Table 2.4 one could once again argue that Classifier 1 performs better than Classifier 2. This is due to the fact that it performs better than Classifier 2 based on One-error and Average Precision, while Classifier 2 only performs better based on Coverage. It is clear that comparing different approaches is more complex in multi-label scenarios than in single-label ones: a given approach may be better than another with respect to some of the evaluation measures, but worse when other measures are considered.

Tsoumakas and Katakis (2007) and Tsoumakas *et al*. (2010) distinguish between two categories for multi-label learning techniques: problem transformation and algorithm adaptation methods. Madjarov *et al*. (2012) add a third category: ensemble methods. A schematic representation of these three categories of techniques is presented in Figure 2.2.

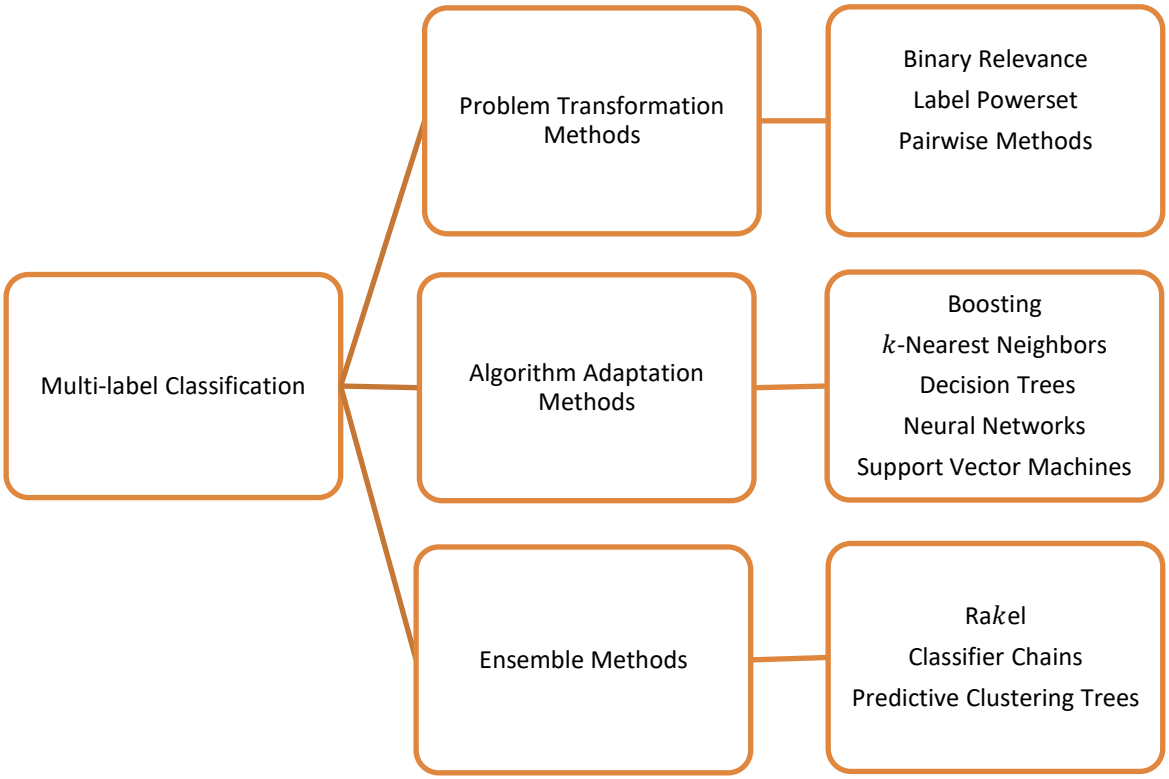


Figure 2.2 Multi-label classification techniques.

These three categories will be discussed in detail in Sections 2.4 to 2.6.

2.4 Problem transformation methods in multi-label classification

In this section, the methods for solving multi-label learning problems by transforming a multi-label dataset to one or more single-label classification dataset(s) are presented. This enables the user to use any of the existing single-label classification algorithms to solve a multi-label classification problem.

Gupta and Anand (2013) argue that problem transformation methods are preferable due to their simplicity and competitiveness across several benchmark datasets. These methods simplify the problem but require that the correlations that exist between the labels are utilised in order to predict new instances accurately.

A discussion of several simple transformations as presented by Tsoumakas *et al*. (2010) that convert a multi-label dataset to a single-label dataset is presented first. These techniques allow

for a ranking to be learned using a single-label classifier. The ranking will be based on the probability of each class. This is followed by a discussion of the three categories of problem transformation methods, namely binary relevance, label powerset, and pairwise methods.

The problem transformation methods below will be illustrated using the multi-label dataset in Table 2.5, as presented by Tsoumakas *et al*. (2010). It consists of four instances that are annotated with one or more of four labels: k_1, k_2, k_3, k_4. As the transformations influence only the label space, the attribute space will be omitted for the sake of simplicity.

Table 2.5: Example of a multi-label dataset.

Example	Attributes	Label set
1	$\mathbf{x_1}$	$\{k_1, k_4\}$
2	$\mathbf{x_2}$	$\{k_3, k_4\}$
3	$\mathbf{x_3}$	$\{k_1\}$
4	$\mathbf{x_4}$	$\{k_2, k_3, k_4\}$

In the following sections, some of the problem transformations discussed in Tsoumakas *et al*. (2010) are presented.

2.4.1 Copy transformation

The *copy transformation* replaces each multi-label instance $(\mathbf{x}_i, \mathbf{Y}_i)$ with $|Y_i|$ instances $(\mathbf{x}_i, k_j)$, for every $k_j \in Y_i$. The dataset resulting from this transformation is given in Table 2.6. It should be noted that in Tables 2.6 and 2.7, each of 1a and 1b will have $\mathbf{x_1}$ as feature vector, and similarly, each of 2a and 2b will have $\mathbf{x}_2$ as feature vector, *etc*.

Table 2.6: Dataset resulting from *copy transformation*.

Example	Label
1a	k_1
1b	k_4
2a	k_3
2b	k_4
3	k_1
4a	k_2
4b	k_3
4c	k_4

Now that the multi-label dataset has been transformed into an ordinary binary dataset containing eight data instances, any binary classification algorithm can be applied. The predictions obtained from the binary classification are then transformed back to multi-label representations. The multi-label representation is obtained by combining the eight binary models, *i.e*. the predicted set of labels is the union of the results predicted from the eight binary models.

A variation of the copy transformation, called the *copy-weight transformation,* links a weight of $\frac{1}{|Y_i|}$ to each of the instances produced by the copy transformation. The dataset resulting from the *copy-weight transformation* is given in Table 2.7.

Table 2.7: Dataset resulting from *copy-weight transformation.*

Example	Label	Weight
1a	k_1	0.50
1b	k_4	0.50
2a	k_3	0.50
2b	k_4	0.50
3	k_1	1.00
4a	k_2	0.33
4b	k_3	0.33
4c	k_4	0.33

In the following section, a discussion of the *select* family of transformations is presented.

2.4.2 Select transformation

The *select-max transformation* replaces $\mathbf{Y}_i$ with the label that occurred most frequently among all the instances. In our example, both k_1 and k_4 are associated with the first instance, but since k_4 occurs more frequently in the complete dataset (three times) than k_1 (twice), k_4 is assigned to instance 1. This type of transformation therefore leads to Table 2.8 below.

Table 2.8: Dataset resulting from *select-max transformation*.

Example	Label
1	k_4
2	k_4
3	k_1
4	k_4

The *select-min transformation* replaces $\mathbf{Y}_i$ with the label that occurred least frequently. See Table 2.9 for the dataset obtained using the *select-min transformation*. There seems to be little motivation for why this transformation would be applied to a dataset.

Table 2.9: Dataset resulting from *select-min transformation*.

Example	Label
1	k_1
2	k_3
3	k_1
4	k_2

The *select-random transformation* replaces $\mathbf{Y}_i$ with a randomly selected label. See Table 2.10 for an example of what such a transformation could potentially look like.

Table 2.10: Dataset resulting from *select-random transformation*.

Example	Label
1	k_1
2	k_4
3	k_1
4	k_3

In the following section, a detailed discussion of the binary relevance transformation is given.

2.4.3 Binary relevance transformation

The *binary relevance* (BR) problem transformation is considered to be the most widely-used approach to multi-label classification. A one-against-all strategy is used to convert the multi-label problem into several single-label classification problems. BR considers the prediction of each label as an independent binary classification task.

Consider the multi-label dataset given in Table 2.5 that contains $N = 4$ observations, $\mathbf{x}_i$, $i = 1,...,4$, and $q = 4$ labels, denoted by Y_j, $j = 1,...,4$. For label k_1 a binary classifier h_1 is trained. Any standard binary classifier, such as logistic regression, random forests, *etc*. can be applied. This implies that the multi-label data are divided into $q = 4$ separate binary classification problems, similar to Table 2.11. The base classifier is trained to each of the $q = 4$ binary classification data and h_k, $k = 1,...,4$, are found.

In general, one binary classifier $h_k : X \to \{\neg k, k\}$ is trained for each of the q different labels $k = 1,...,q$. There are q binary classifiers in total; one for each label. The original dataset is correspondingly transformed into q binary datasets that contain all examples of the original dataset, labelled as k if the labels of the original instance included k and as $\neg k$ if not.

For the classification of a new instance, BR outputs the union of the labels k_j that are positively predicted by the q binary classifiers. For the data in Table 2.5, this would imply that an unseen observation $\mathbf{x}$ is classified by $\left[h_1(\mathbf{x}) \quad h_2(\mathbf{x}) \quad h_3(\mathbf{x}) \quad h_4(\mathbf{x})\right]$.

Table 2.11 below shows the four datasets that are constructed by BR when applied to the example dataset.

Table 2.11: Four BR datasets.

Dataset 1	
Example	**Label**
1	k_1
2	$\neg k_1$
3	k_1
4	$\neg k_1$

Dataset 2	
Example	**Label**
1	$\neg k_2$
2	$\neg k_2$
3	$\neg k_2$
4	k_2

Dataset 3	
Example	**Label**
1	$\neg k_3$
2	k_3
3	$\neg k_3$
4	k_3

Dataset 4	
Example	**Label**
1	k_4
2	k_4
3	$\neg k_4$
4	k_4

There are two advantages to the BR approach. Firstly, it has a low computational complexity when compared with other approaches. Secondly, the approach allows for any of the binary classifiers to be applied.

The BR approach has two key shortcomings, namely:

1. For a large number of labels, the BR approach may experience drawbacks from the imbalanced data problem. The reason is that it is more likely that for some binary classifiers, the number of negative examples that indicate that the specific label is not present, could be much larger than the number of positive examples. As a result, some binary classifiers might predict negative for all new instances.
2. This method does not consider the possible correlations between the different labels. Cherman *et al*. (2010) propose a simple approach named BR+, which can be employed to incorporate label dependency aiming to accurately predict label combinations.

Due to the critical information contained in the label dependencies, it is beneficial to take these dependencies into account. Due to the exponentially expanding number of possible

combinations of labels as $|Y_i|$ increases, methods which model all label dependencies will have very high complexity. If one considers the low complexity of the BR method, it could be useful to adapt it to incorporate the label independencies.

BR+ can be used to incorporate label dependency aiming to accurately predict label combinations (Cherman *et al.*, 2010). This approach does not attempt to detect existing label dependency before a classifier is introduced. The aim is to allow classifiers to detect these label dependencies automatically.

During the training phase, BR+ functions in a similar manner to BR, namely that q binary classifiers are generated for each label. BR+ increments the feature space with $q-1$ features, which correspond to other labels in the multi-label dataset (Cherman *et al.*, 2012). Each one of the binary training instances is therefore augmented with ϖ_j binary features where $\varpi_j = Y_i - \{k_j\}$.

BR+ allows the user to specify different ways of considering the labels in order to explore label dependency, where three of them have been implemented in BR+ (Cherman *et al.*, 2010). The first prediction strategy is called *No Update (NU)*, since no modification is made to the original estimates of the augmented features during the prediction phase. If a predefined order to predict the individual labels is considered, each of the new values is used to update the previous value (the initial prediction) of the corresponding augmented features of the unlabelled example. This prediction strategy is called *Static Order (Stat)*. The last prediction strategy is called *Dynamic Order (Dyn)*. This strategy predicts and updates labels with less confidence first during the final prediction phase of BR+. See Cherman *et al.* (2010) and Cherman *et al*. (2012) for more details on BR+.

Empirical evaluation on multi-label datasets from different domains was performed by Cherman *et al.* (2012). Initial results show that BR+ has the potential to improve the multi-label classification performance in datasets with low dimensionality in the labels. However, these results are not statistically significant, and the authors propose that more extensive empirical evaluation should be performed.

In the following section, the label powerset approach will be discussed in detail based on the same example from Tsoumakas *et al*. (2010).

2.4.4 Label powerset transformation

The *label powerset* (LP) transformation creates a unique label for each combination of labels that exists in a multi-label training set and considers these unique labels as classes in a new single-label multi-class classification task. Table 2.12 presents the result of transforming the dataset of Table 2.5 using the LP approach.

Table 2.12: Example of an LP multi-label dataset.

Example	Label
1	$k_{1,4}$
2	$k_{3,4}$
3	k_1
4	$k_{2,3,4}$

This technique attempts to take the correlations between labels into consideration. For a new instance the single-label multi-class classifier of LP provides the class with the largest probability. This class represents a set of labels. If the single-label classifier is able to provide posterior probabilities, the LP transformation can also provide a ranking of the labels. A label ranking can then be obtained by summing the probabilities of the classes that contain the specific label. Table 2.13 shows an example of a probability distribution that can be produced by LP, trained on the dataset in Table 2.12. This example is taken from Tsoumakas *et al.* (2010).

Table 2.13: Example of a ranking obtained using LP.

c	$p(c \mid \mathbf{x})$	k_1	k_2	k_3	k_4
$k_{1,4}$	0.7	1	0	0	1
$k_{3,4}$	0.2	0	0	1	1
k_1	0.1	1	0	0	0
$k_{2,3,4}$	0.0	0	1	1	1
	$\sum_c p(c \mid \mathbf{x}) k_j$	0.8	0.0	0.2	0.9

The posterior probability for each set of labels in Table 2.13 would be obtained by the classifier. For a new case, the label ranking is calculated by summing the probability of each of the classes which are present in the label set. For example, consider the third column; the relevant probabilities in the first column are summed as follows: 1(0.7) + 0(0.2) + 1(0.1) + 0(0.0) = 0.8. This calculation is done for each class, which gives the total probability associated with each label. According to this approach label 4 is ranked first with a probability of 0.9, followed by labels 1, 3, and 2. An arbitrary threshold value is then used to partition between relevant and irrelevant labels.

The LP technique has two significant limitations:

a) It can only classify a new, unseen instance to a label set present in the training dataset. This implies that new label combinations not present in the training dataset cannot be formed.
b) LP may lead to datasets with a large number of classes $\left(2^{q-1}\right)$ with very few examples in some classes. This is especially true if q is large, *i.e.* if there are a large number of labels. This aspect could have an adverse effect on the computation time of the classifier.

Read *et al*. (2008) proposes the method of pruned problem transformations to address these limitations. The method only includes the distinct label sets which occur more than a predefined number of times, φ. Label sets which occur fewer than φ times are discarded, and training takes place on the pruned datasets. The random k-labelsets (RAkEL) method also attempts to address some of these drawbacks while still leveraging the same fundamental concept as LP. A detailed discussion of RAkEL will be provided in Section 2.6.

The method of ranking by pairwise comparison is discussed in the next section.

2.4.5 Pairwise methods

Ranking by pairwise comparison (RPC) transforms the multi-label dataset into $\frac{q(q-1)}{2}$ binary label datasets, one for each pair of labels $(k_i, k_j), 1 \le i < j \le q$. Each dataset contains those instances that are indexed by exactly one of the two corresponding labels. A binary classifier that learns to discriminate between the two labels from each of these datasets is then trained.

A new instance is classified using all $\frac{q(q-1)}{2}$ binary classifiers and a ranking is obtained by counting the number of votes received by each label. Applying RPC to the example dataset, the datasets presented in Table 2.14 are obtained, and a binary classifier is fitted for each one of these six datasets.

Table 2.14: Six RPC datasets.

Dataset 1: (1,2)	
Example	**Label**
1	$k_{1,\neg 2}$
3	$k_{1,\neg 2}$
4	$k_{\neg 1,2}$

Dataset 2: (1,3)	
Example	**Label**
1	$k_{1,\neg 3}$
2	$k_{\neg 1,3}$
3	$k_{1,\neg 3}$
4	$k_{\neg 1,3}$

Dataset 3: (1,4)	
Example	**Label**
2	$k_{\neg 1,4}$
3	$k_{1,\neg 4}$
4	$k_{\neg 1,4}$

Dataset 4: (2,3)	
Example	**Label**
2	$k_{\neg 2,3}$

Dataset 5: (2,4)	
Example	**Label**
1	$k_{\neg 2,4}$
2	$k_{\neg 2,4}$

Dataset 6: (3,4)	
Example	**Label**
1	$k_{\neg 3,4}$

A new instance is classified by obtaining a ranking by counting the votes received by each label for each binary classifier constructed. This ranking requires a threshold that will split the labels between those that are relevant and those that are not.

RPC is extended by the method of calibrated ranking. For more information on calibrated label ranking see Tsoumakas *et al*. (2010) and Fürnkranz *et al*. (2008).

A number of single-label classifiers have been modified to directly solve multi-label problems. The following section will briefly mention some of these algorithm adaptation methods.

2.5 Algorithm adaptation methods in multi-label classification

Existing single-label classification algorithms can be extended to deal with multi-label data directly. Examples of such methods include neural networks (Crammer and Singer, 2003 and Zhang and Zhou, 2006), boosting (Schapire and Singer, 2000 and De Comité *et al*., 2003), classification rules (Thabtah *et al*., 2004), decision trees (Clare and King, 2001 and Blockeel

et al., 1998), lazy learning (Zhang and Zhou, 2007, Wieczorkowska *et al*., 2006, and Spyromitros *et al*., 2008).

In this section, the discussion of algorithm adaptation methods is limited to multi-label k-nearest neighbours.

2.5.1 Multi-label k-Nearest Neigbours (ML-kNN)

The most widely used multi-label variation of kNN is the approach by Zhang and Zhou (2007), namely ML-kNN. The following discussion is based on this paper.

In ML-kNN – exactly as in die single-label classification problem – the first step is to calculate the nearest neighbours of the instance to be classified. The frequency of each label among the nearest neighbours is used to estimate prior and posterior probabilities. Based on these prior and posterior probabilities the maximum a posteriori principle is used to then determine the label set of an unseen instance.

Consider the situation where one wishes to apply 5-NN in a multi-label problem. Consider a new, unseen instance, $\mathbf{x}$, with $N_5(\mathbf{x})$ its set of nearest neighbours in the training dataset. These are the five data instances $(\mathbf{x}_i, \mathbf{y}_i)$ with input vectors closest to $\mathbf{x}$. In Table 2.15 below, an illustrative dataset with three labels, $q = 3$ is presented.

Table 2.15: Example of a nearest neighbour dataset

Nearest neighbours of x	**Labels**		
	Y_1	Y_2	Y_3
$\mathbf{x}_1$	1	1	0
$\mathbf{x}_2$	0	1	1
$\mathbf{x}_3$	1	0	0
$\mathbf{x}_4$	0	1	1
$\mathbf{x}_5$	1	1	0
	$T_{\mathbf{x}}(1) = 3$	$T_{\mathbf{x}}(2) = 4$	$T_{\mathbf{x}}(3) = 2$

One can compute the label counting totals, $T_{\mathbf{x}}(\bullet)$ for $N_5(\mathbf{x})$ as in the last row of Table 2.15. For $k \in \{1,2,...,q\}$, one has $T_{\mathbf{x}}(k) \in \{0,1,...,5\}$.

For every label $k \in \{1,2,...,q\}$, define $H_1^k(\mathbf{x})$ as the event that $\mathbf{x}$ has $Y_k = 1$, $H_0^k(\mathbf{x})$ as the event that $\mathbf{x}$ has $Y_k = 0$, and E_m^k as the event that $T_{\mathbf{x}}(k) = m$, $m \in \{0,1,...,5\}$. A Bayesian approach is followed to assign labels to $\mathbf{x}$. The maximum a posteriori (MAP) probability gives $Y_k(\mathbf{x}) = \arg\max_{b \in \{0,1\}} P\left(H_b^k \mid E_{T_{\mathbf{x}}(k)}^k\right)$, $k = 1,2,...,q$. This implies that in order to decide whether $Y_k(\mathbf{x}) = 1$ or $Y_k(\mathbf{x}) = 0$, the condition on $E_{T_{\mathbf{x}}(k)}^k$ is taken. Applying Bayes' Theorem, $Y_k(\mathbf{x})$ becomes

$$\arg\max_{b \in \{0,1\}} \left\{ P\left(H_b^k\right) P\left(E_{T_{\mathbf{x}}(k)}^k \mid H_b^k\right) \right\}, \quad k = 1,2,...,q .$$

The prior, $P\left(H_b^k\right)$, and the likelihood, $P\left(E_{T_{\mathbf{x}}(k)}^k \mid H_b^k\right)$, have to be specified. The prior probabilities are estimated using an empirical Bayes approach:

$$\hat{P}\left(H_1^k\right) = \frac{s + \sum_{i=1}^{N} Y_{ik}}{2s + N} \text{ and } \hat{P}\left(H_0^k\right) = 1 - \hat{P}\left(H_1^k\right),$$

where s is a smoothing parameter which controls the strength of the uniform prior assumption. $P\left(E_{T_{\mathbf{x}}(k)}^k \mid H_1^k\right)$ is the conditional probability of observing $T_{\mathbf{x}}(k)$ nearest neighbours of $\mathbf{x}$ having $Y_k = 1$, given that $\mathbf{x}$ has $Y_k = 1$. Similarly, $P\left(E_{T_{\mathbf{x}}(k)}^k \mid H_0^k\right)$ is the conditional probability of observing nearest neighbours of $\mathbf{x}$ having $Y_k = 1$, given that $\mathbf{x}$ has $Y_k = 0$.

The procedure for estimating $P\left(E_{T_{\mathbf{x}}(k)}^k \mid H_1^k\right)$ for each $m \in \{0,1,...,5\}$ is as follows:

1) Consider all $\sum_{i=1}^{N} Y_{ik}$ cases in the sample data with $Y_k = 1$.

2) Calculate a vector $\mathbf{W}$ of totals as follows: $\mathbf{W}[m]$ = the number of times (out of $\sum_{i=1}^{N} Y_{ik}$) where exactly m of the 5 nearest neighbours had $Y_k = 1$.

3) Now estimate $P\left(E^k_{T_x(k)} \mid H^k_1\right)$ by $\hat{P}\left(E^k_m \mid H^k_1\right) = \dfrac{s + \mathbf{W}[m]}{6s + \sum_{r=0}^{5} \mathbf{W}[r]}$.

$P\left(E^k_{T_x(k)} \mid H^k_0\right)$ can be estimated by following a similar procedure for each $m \in \{0,1,...,5\}$. The label vector for a new instance can now be predicted using

$$\hat{Y}_k(\mathbf{x}) = \arg\max_{b \in \{0,1\}} \left\{ \hat{P}\left(H^k_b\right) \hat{P}\left(E^k_{T_x(k)} \mid H^k_b\right) \right\}, \quad k = 1,2,...,q .$$

In the comparative study performed by Madjarov *et al*. (2012), the ML-kNN approach does not perform particularly well when compared to other algorithm adaptation techniques. For alternative lazy learning algorithms, refer to Wieczorkowska *et al*. (2006) and Spyromitros *et al*. (2008). Spyromitros *et al*. (2008) suggest that the method that they propose, namely a combination of BR and kNN – BRkNN – performs better than the ML-kNN method.

In the following section, ensemble methods for multi-label classification problems will be discussed.

2.6 Ensemble methods in multi-label classification

These methods are developed on top of either problem transformation or algorithm adaptation methods. Random k-labelsets (RAkEL) (Tsoumakas and Vlahavas, 2007), ensembles of pruned sets (Read *et al*., 2008), and ensembles of classifier chains (Read *et al*., 2009) are all examples of problem transformation ensembles. Ensembles of predictive clustering trees (Dimitrovski *et al*., 2012) is an example of an algorithm adaptation ensemble.

RAkEL, proposed by Tsoumakas and Vlahavas (2007), is an ensemble of label powerset (LP) classifiers. Each LP classifier is trained using a different small random subset of the full set of labels. The proposed method aims to take into account label correlations while not suffering from the large number of label subsets with the majority associated with very few examples.

Ensemble combination is accomplished by thresholding the average zero-one decisions of each model per considered label.

A set $Y \subseteq L$ with $k = |Y|$ is called a k-labelset. Tsoumakas and Vlahavas (2007) use the symbol L^k to denote the set of all distinct k-labelsets in L. The size of L^k is given by the binomial coefficient: $|L^k| = \binom{q}{k}$. The algorithm iteratively constructs an ensemble of m LP classifiers. At each iteration, $i = 1, 2, ..., m$, the algorithm randomly selects a k-labelset from L^k without replacement.

The number of iterations, m is a parameter specified by the user with a range from 1 to $|L^k|$. The size of the labelsets, k is also specified by the user with a meaningful range between 2 and $|L^k| - 1$. If $k = 1$ and $m = |L^k|$, the binary classifier ensemble of BR is obtained, while if $k = |L^k|$ (and accordingly $m = 1$) the single label classifier of LP is obtained. Tsoumakas and Vlahavas (2007) theorise that when using small k-labelsets with an adequate number of iterations, RAkEL will model label correlations effectively.

For multi-label classification of a new instance $\mathbf{x}$, each model h_i provides binary decisions for each label in the corresponding k-labelset Y_i. The RAkEL algorithm calculates the average decision for each label k_j in L^k and outputs a final positive decision if the average is greater than a user-specified threshold t. As with LP, a threshold value of 0.5 is used. The empirical results show that RAkEL performs well across a wide range of values of t.

One important feature of RAkEL is the high number of class values, 2^k, that each LP classifier must learn. This could become an important limitation of the proposed algorithm, especially if the base classifier has quadratic or greater complexity with respect to the number of class values, as in the case of support vector machine classifiers. In practice, the true number of class values is never 2^k, because LP can only consider the label subsets that appear in the training data. Typically, the number of these subsets is significantly smaller than 2^k.

In the comparative study by Madjarov *et al*. (2012), RAkEL performs relatively poorly compared to the other methods.

In Section 2.7, the base classifiers used in a BR approach in this study are described in detail.

2.7 Base classifiers

The two algorithms/classifiers used for classification in this dissertation are support vector machines (SVMs) and extreme gradient boosting (XGBoost). This section provides an overview of each of these classifiers.

2.7.1 Support vector machines

This section proceeds with a description of a popular machine learning approach to classification problems, *viz*. the support vector machine (SVM). This is not intended to be a detailed description, but rather a brief overview of the topic. For more in-depth discussions, see for example Schölkopf and Smola (2002) and Hastie *et al*. (2009).

Consider a binary classification dataset, $D = \{(\mathbf{x}_i, y_i), i = 1,..., N\}$, where each $\mathbf{x}_i$ is a p-component vector of inputs, and $y_i \in \{-1, +1\}$ denotes its corresponding label. Let $f : \mathbb{R}^p \to \mathbb{R}$ be a discriminant function with $sign[f(\mathbf{x})]$ being the class assigned to the input vector $\mathbf{x}$. A scenario where classification is fairly straightforward is that where the data are linearly separable. In this case, one can find a hyperplane (an extension of the concept of a line for $p > 2$ dimensions) that perfectly separates the two groups in D. More specifically, one can find an intercept β_0 and a slope vector $\boldsymbol{\beta}$ so that $sign[\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle] = y_i$ for all $i = 1,..., N$. In this expression, $\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle = \sum_{j=1}^{p} \beta_j x_{ij}$ is the usual inner product between two vectors.

Assuming linearly separable data D, how does one go about finding β_0 and $\boldsymbol{\beta}$? The perceptron (see for example Rosenblatt, 1958) achieves this by iteratively decreasing the total distance of misclassified points from the hyperplane. The optimal separating hyperplane (the simplest example of an SVM) approaches the problem as follows. Consider a separating hyperplane $H(\beta_0, \boldsymbol{\beta}) \equiv H = \{\mathbf{x} : \beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle = 0\}$. Since H is a separating hyperplane,

$$y_i (\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle) \geq 0, \text{ for all } i = 1,..., N.$$

The margin of H with respect to . D . is the minimum distance that any point in D lies from H . Since the (signed) distance between a point $\mathbf{x}$ and the hyperplane H is given by

$$\frac{\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle}{\|\boldsymbol{\beta}\|},$$

it follows that the margin of H with respect to D is given by

$$\min\left\{ \frac{y_i\left[\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right]}{\|\boldsymbol{\beta}\|} : \mathbf{x}_i \in D \right\}.$$

The optimal separating hyperplane (OSH) is defined to be the hyperplane with maximal margin.

It can be argued that (see for example the references provided earlier) the OSH is the solution to the optimization problem

$$\min_{\beta_0, \boldsymbol{\beta}} \left\{ \frac{1}{2}\|\boldsymbol{\beta}\|^2 \right\}$$

subject to the constraints $y_i\left(\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right) \geq 1$, $i = 1, ..., N$. The constraints are taken into consideration by introducing non-negative Lagrange multipliers, $\alpha_1, ..., \alpha_N$, and the Lagrangian objective function becomes

$$L(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i\left(\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right) - 1 \right]. \tag{2.1}$$

This function has to be minimised with respect to $\boldsymbol{\beta}$ and β_0, and maximised with respect to $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_N \end{bmatrix}^T$. A standard approach to accomplish this, proceeds as follows. The partial derivatives of L with respect to $\boldsymbol{\beta}$ and β_0 are given by

$$\frac{\partial L(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha})}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

and

$$\frac{\partial L(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha})}{\partial \beta_0} = \sum_{i=1}^{N} \alpha_i y_i .$$

Setting these derivatives equal to zero gives

$$\boldsymbol{\beta} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{2.2}$$

and

$$\sum_{i=1}^{N} \alpha_i y_i = 0 . \tag{2.3}$$

If (2.2) and (2.3) are substituted into (2.1), the dual form of the optimisation problem is obtained, *viz*. maximise

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{2.4}$$

subject to $\alpha_i \geq 0,\ i = 1, ..., N$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$. The dual optimisation problem can easily be solved using quadratic optimisation, thereby obtaining the optimal values $\hat{\alpha}_1, ..., \hat{\alpha}_N$.

It follows from (2.2) that the slope vector of the OSH is given by

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} \hat{\alpha}_i y_i \mathbf{x}_i , \tag{2.5}$$

a linear function of $\mathbf{x}_1, ..., \mathbf{x}_N$. From the Karush-Kuhn-Tucker conditions for the dual optimisation problem (these conditions are not discussed here), it follows that in many cases a sizeable number of the $\hat{\alpha}_i$'s equal zero. Let V be the set of indices for which $\hat{\alpha}_i > 0$, *i.e.* $V = \{i : \hat{\alpha}_i > 0\}$. Then (2.5) becomes

$$\hat{\boldsymbol{\beta}} = \sum_{i \in V} \hat{\alpha}_i y_i \mathbf{x}_i . \tag{2.6}$$

The inputs $\mathbf{x}_i$, $i \in V$, are the support vectors of the OSH: once $\hat{\alpha}_i$, $i = 1,...,N$, have been determined, the OSH depends only on the support vectors. In this sense the OSH (and later also the SVM) has a sparsity property.

Regarding the optimal value of the intercept, $\hat{\beta}_0$, an argument based on the form of the Karush-Kuhn-Tucker conditions implies that $\hat{\beta}_0$ can be calculated from

$$\hat{\beta}_0 = y_i - \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle, \tag{2.7}$$

for any $i \in V$. For numeric stability, $\hat{\beta}_0$ is usually taken equal to the mean of the values obtained from (2.7) for all the support vectors.

Two remarks conclude this discussion of the OSH. Firstly, the OSH classifier assigns the class

$$sign\left[\beta_0 + \langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right] = sign\left[\hat{\beta}_0 + \sum_{i \in V} \hat{\alpha}_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle\right] \tag{2.8}$$

to a data case with feature vector $\mathbf{x}$. Secondly, from (2.4), (2.7), and (2.8) it can be argued that the OSH depends on the input vectors only through inner products. More specifically, once the matrix

$$G : N \times N = \left[\langle \mathbf{x}_i, \mathbf{x}_j \rangle\right], i, j = 1,...,N \tag{2.9}$$

has been computed, the individual vectors $\mathbf{x}_1,...,\mathbf{x}_N$ can be discarded. Similarly, given a new input vector $\mathbf{x}$, it is only the inner products $\langle \mathbf{x}_i, \mathbf{x} \rangle, i \in V$, that will determine the classification of $\mathbf{x}$.

The support vector classifier (SVC) and the SVM arise as extensions of the OSH to cases where the training data are not linearly separable and where non-linear decision boundaries may be required. There are two basic ideas that play an important role in these extensions: firstly, the introduction of slack variables to provide for the fact that the training data are not linearly separable; secondly, replacing every inner product in the OSH by a corresponding kernel function $K(\bullet,\bullet)$ evaluated on the pair of inputs involved in the inner product.

Omitting details regarding the required derivations, the general SVM classifier is given by

$$sign\left[\hat{\beta}_0 + \sum_{i \in V} \hat{\alpha}_i y_i K\left(\mathbf{x}_i, \mathbf{x}\right)\right]. \tag{2.10}$$

In this expression, $\hat{\beta}_0$ and $\hat{\alpha}_1, ..., \hat{\alpha}_N$ are once again determined from D by maximising the dual form of an optimisation problem similar to the one in (2.4). An important difference, though, is that every inner product $\left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle$ between two input vectors is replaced by a kernel function evaluation $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$.

A kernel function is a symmetric, positive semi-definite function of two p-component arguments. The radial basis (Gaussian) kernel, given by

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = e^{-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2}, \tag{2.11}$$

is a popular choice and often performs well. In this expression the quantity γ is a kernel hyperparameter, usually determined by means of cross-validation.

Two remarks conclude this discussion. Firstly, the SVM, as is the case for the OSH, has the sparsity property. In (2.10) the summation therefore only involves the set of support vectors which, in some cases, will only be a small subset of all the input vectors. Secondly, whereas the OSH is a linear classifier, this is not the case for the SVM. Apart from a very specific choice of the kernel function, the expression in (2.10) will be a non-linear function of $\mathbf{x}$.

In the next section, a discussion of boosting, with specific reference to XGBoost, is presented.

2.7.2 Extreme gradient boosting

According to Hastie *et al*. (2009) boosting is considered to be one of the most powerful machine learning techniques introduced into the field in the last 20 years. Boosting is an ensemble method that aims to build a *strong* model based on *weak* classifiers. The terms *strong* and *weak* here refer to how correlated the learners are to the target. Models are added "on top" of each other in an iterative manner, which allows for the errors of the previous model to be corrected by the next classifier until the training data are accurately predicted by the final ensemble model.

A discussion of *boosting* will not be complete without some brief comments about *bagging* first. Bagging or *bootstrap aggregation* averages a given procedure over many samples in order to reduce its variance (Hastie *et al*., 2009). If the base classifier is a decision tree, bagging fits many trees to bootstrap samples selected from the training data, and then performs classification by means of a majority vote.

Suppose $H(\mathbf{X},\mathbf{x})$ is a classifier, such as a tree, producing a predicted class label for input $\mathbf{x}$. To bag H, one draws bootstrap samples $\mathbf{X}^{*1},...,\mathbf{X}^{*B}$ each of size N with replacement from the training dataset. Then $\hat{H}_{bag}(\mathbf{x}) = \text{Majority Vote}\left\{H(\mathbf{X}^{*b},\mathbf{x})\right\}_{b=1}^{B}$. This means that the overall prediction is the class which occurs most commonly among the B predictors.

Bagging can dramatically reduce the variance of unstable procedures which will lead to more accurate prediction.

The method of boosting is based on the following two modifications, namely, using a weighted sample instead of using a random sample from the training dataset, and using a weighted vote when combining classifiers. The first modification focuses learning on the examples which, at any stage of the process, are most difficult to classify.

Schapire (1990) and Freund (1995) developed earlier methods, but this dissertation will focus on the method *AdaBoost* as proposed by Freund and Shapire (1997). Once again, consider a binary classification dataset, $D = \left\{(\mathbf{x}_i, y_i), i = 1,...,N\right\}$, where each $\mathbf{x}_i$ is a p-component vector of inputs, and $y_i \in \{-1,+1\}$ denotes its corresponding label. A classifier $H(\mathbf{x})$ produces a prediction taking on either a value of –1 or +1. The error rate on the training sample is

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} I\left(y_i \neq H(\mathbf{x}_i)\right),$$

with the expected error being $E_{XY}I\left(Y \neq H(\mathbf{x})\right)$ on new or unseen input vectors. A weak classifier is a classifier with an error rate only slightly better than random guessing. The aim of boosting is to sequentially apply a weak classifier to modified versions of the data. This produces a sequence of weak classifiers, $H_m(\mathbf{x})$, $m = 1,...,M$. The M predictions are combined through a weighted majority vote, producing the final prediction

$$H(\mathbf{x}) = sign\left(\sum_{m=1}^{M} \alpha_m H_m(\mathbf{x})\right),$$

where $\alpha_1,...,\alpha_M$ are calculated by the boosting algorithm. These values determine the weight of each respective $H_m(\mathbf{x})$, which gives greater influence to the classifiers that are more accurate. Figure 2.3 shows a schematic representation of the AdaBoost classifier. This figure has been adapted from Gandhi (2018) and Hastie *et al*. (2009).

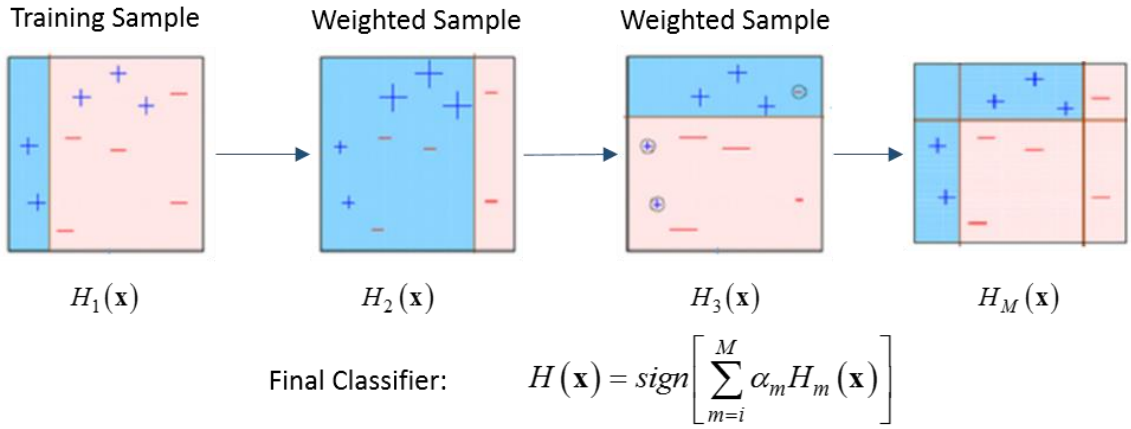


Figure 2.3 Schematic representation of the AdaBoost algorithm.

The weighted samples at each boosting step are obtained by applying weights $w_1,...,w_N$ to each of the training observations. At the first boosting step, where $m = 1$, all these weights are set to $w_i = 1/N$. For each of the subsequent boosting steps ($m = 2,...,M$), the weights are modified individually. At any boosting step m, the weights for the observations that were misclassified by the classifier $H_{m-1}(\mathbf{x})$ in the previous step, are increased, while the weights for those observations that were classified correctly are decreased. As the number of boosting steps increases, observations which are difficult to classify correctly become more influential as their weights increase. This allows a classifier at step m to concentrate on those training observations that had been missed by classifiers in the previous steps.

The method can be summarised as follows:

- Initialise the observation weights for the training dataset, $w_i = 1/N$, $i = 1,...,N$.
- For $m = 1,...,M$:
    a) Train a weighted weak learner $H_m(\mathbf{x})$ to the training dataset using the weights w_i.
    b) Calculate the weighted training error rate, $\text{err}_m = \dfrac{\sum_{i=1}^{N} w_i I\left(y_i \neq H(\mathbf{x}_i)\right)}{\sum_{i=1}^{N} w_i}$.
    c) Calculate $\alpha_m = \log\left(\left(1-\text{err}_m\right)/\text{err}_m\right)$.
    d) Set the weight $w_i \leftarrow w_i \cdot e^{\alpha_m I\left(y_i \neq H_m(\mathbf{x}_i)\right)}$, $i = 1,...,N$.

The final classifier is a weighted sum $H(\mathbf{x}) = sign\left(\sum_{m=1}^{M} \alpha_m H_m(\mathbf{x})\right)$.

The algorithm above is referred to as *AdaBoost.M1* or *Discrete AdaBoost* because a discrete class label is predicted. If a real-valued prediction is returned by the base classifier, the algorithm is considered to be *Real AdaBoost*. Refer to Friedman *et al*. (2000) for more details.

Gradient boosting functions in a similar manner, but instead of assigning different weights to the observations after each iteration, this ensemble method fits the model to the residuals from the previous prediction and then minimises the loss when adding the next classifier. Specific algorithms are obtained by utilising different loss functions $L\left(y, f\left(\mathbf{x}\right)\right)$. The model is updated using gradient descent. For an in-depth discussion see Hastie *et al*. (2009) and Bühlmann and Hothorn (2007).

XGBoost is a flexible and versatile tool that is a scalable and accurate implementation of gradient boosting machines (Chen and Guestrin, 2016). It implements this algorithm for decision tree boosting with an extra custom regularisation term in the objective function. The popularity of XGBoost is apparent if one considers the machine learning competitions hosted by Kaggle (Brownlee, 2016). For example, among the 29 challenge winning solutions published by Kaggle in 2015, 17 solutions made use of XGBoost.

2.8 Conclusion

In the first section of this chapter, the important concepts of multi-label classification, such as the aim of multi-label classification, the unique characteristics of multi-label datasets, and benchmark multi-label datasets, were discussed in detail. Secondly, the evaluation measures pertaining to multi-label classification techniques were presented. Specific mention was made of the example-based evaluation measures and the measures based on label rankings.

In Sections 2.4 to 2.6 the relevant research around the topics of problem transformation methods, algorithm adaptation methods, and ensemble methods were presented. Detailed discussions of BR, LP, and RPC were provided in Section 2.4. In Section 2.5, the method of ML-kNN was presented and in Section 2.6, the ensemble technique, RAkEL was discussed. Finally, the two base classifiers used in this study, namely SVMs and XGBoost, were described.

In Chapter 3, the field of feature selection in the multi-label context will be examined.

CHAPTER 3

MULTI-LABEL FEATURE SELECTION: EXISTING METHODS AND A NEW PROPOSAL

3.1 Introduction

Due to the tremendous increase in the ease and cost of collecting and storing data, which can be ascribed to the increase in computing capacity over the last ten years, feature selection (FS) has become increasingly important in the field of machine learning. Datasets from applications such as genomics, text categorisation, and computational biology are known to be characterised as *wide* datasets, *i.e.* datasets where the number of features are much larger than the number of instances, $p >> N$.

As mentioned in Section 1.2, during FS the feature space $X = \{X_1, X_2, ..., X_p\}$ is searched to find a subset $X' \subseteq \mathrm{X}$ such that X' describes the dataset almost as well as X does. As such, the objectives of FS are:

1. removal of irrelevant and/or redundant features;
2. improved prediction performance of the classifier;
3. improved classification speed and cost efficacy; and
4. to better understand the underlying process that generated the data.

Regarding the first objective above, the removal of irrelevant and/or redundant features is a powerful tool. *Irrelevant features* are features that do not provide any information about the classification problem. The removal of an irrelevant feature will therefore not have a negative influence on performance as information is not lost due to the elimination of such features. *Redundant features* do contain information which is of value to the classification problem, but these features provide the same information as one or more of the other features in the dataset. Since the information is “duplicated”, these redundant features can be eliminated without having a negative influence on performance as no unique information is lost after it has been removed. The success of an FS method lies in its ability to identify and remove irrelevant and redundant features from the dataset.

The inclusion of irrelevant and/or redundant features in a classification dataset introduces unnecessary noise which will have a negative impact on the performance of a classifier. If the dimensionality is reduced, it will lead to faster learning algorithms and – in some cases – even to improved performance. A number of studies (Spolaôr *et al*., 2013, Trohidis *et al*., 2008, and Zhao *et al*., 2011) have been conducted which show that FS can be performed without reduced performance. Dendamrongvit *et al*. (2011) show that the performance of the induced classifiers is impaired by the presence of a large number of irrelevant features. The authors consider two machine learning techniques, namely nearest-neighbour classifiers and SVMs and show that both these techniques benefit from the proposed FS methods. The benefits are in the form of more balanced values for Precision and Recall.

When FS leads to a large reduction of the feature space, the third benefit or objective – of improving classification speed and cost efficacy – is of particular importance. Using fewer features during the classification process will lead to improvements in the processing speed of the algorithm and reduce costs.

The final objective of FS is to provide a better understanding of the data. This is of great benefit as information regarding which features are relevant to the classification task can assist greatly with the interpretability of the classification problem. In medical research, such as that of Cheng *et al*. (2015) and Cheng *et al*. (2018), for example, FS allows the researcher to gain understanding of the importance of each of the features. This means that the importance of features which are collected from expensive sources (such as magnetic resonance imaging (MRI) scans) can be determined.

In this chapter, a brief summary of the general approaches to FS will be presented in Section 3.2. Specific mention will be made of feature ranking and feature subset selection. Existing multi-label FS methods will be examined in Section 3.3. This section will include a discussion of the importance of FS in the multi-label context as well as a detailed description of the relevance measures employed in this dissertation. Finally, the methods of Probe Selection (Sandrock and Steel, 2016) and the method proposed by Spolaôr *et al*. (2013) will be discussed in detail in this section. In Section 3.4, a new multi-label FS method, relevance pattern feature selection (RPFS) is proposed. This section will include some brief background on both biplots and the field of CA. Finally, a summary will be provided in Section 3.5.

3.2 Approaches to feature selection

FS algorithms evaluate the suitability of features from two main perspectives, namely individual and subset evaluation. Individual evaluation is computationally less expensive as it evaluates individual features and assigns ranks/weights to each according to their degree of class prediction. A shortcoming of this method is that it is incapable of identifying irrelevant and/or redundant features since they will all have similar ranks (Spolaôr *et al*., 2013).

The subset evaluation method can deal with both feature relevance and feature redundancy. For subset evaluation, the evaluation measures are defined against a subset of features, leading to higher computational complexity.

The approaches to FS discussed in this section are limited to the single-label problem.

3.2.1 Feature ranking

Various FS algorithms include feature ranking as an initial selection process. Feature ranking is popular due to its simplicity, scalability, and empirical performance. A ranking criterion is obtained which can distinguish between for example healthy and diseased patients.

Consider a training dataset $\{(\mathbf{x}_i, y), i = 1, 2, ..., N\}$ consisting of N observations on p input features, $X_1, X_2, ..., X_p$, and corresponding observations on a single categorical response variable Y. Feature ranking utilises a scoring function that is computed from $\mathbf{x}_i$ and y_i, $i = 1, 2, ..., N$. Typically, one assumes that a high score indicates that a feature adds more value than a feature with a low score. Features are ranked in decreasing order of the scoring function. Subsets can now be built by including more and more features until a subset of optimal size is found. For a complete discussion on the selection of an optimum subset size, refer to Guyon and Elisseeff (2003).

Quevedo *et al*. (2007) devise an algorithm to rank input features using a combination of techniques, specifically correlation and orthogonalisation, which allows for the efficient ranking of features. Some proposed ranking techniques rely on a learning algorithm to rank features, for example kernel-based methods such as SVMs. The SVM-based ranking methods employ an iterative approach where one feature, which is deemed the least useful, is ruled out at each step (Rakotomamonjy, 2003). Another popular ranking criterion is based on mutual

information which has been implemented in the R library `varrank` and is discussed in more detail by Krier *et al*. (2006)

Feature ranking has received criticism because it could lead to the selection of a redundant subset. The approach fails to answer the crucial question: Can a smaller subset of features that could achieve similar performance be obtained?

The ranking of individual features according to their predictive power does not adequately address the task at hand. In the next section, feature subset selection will be examined as a potential solution.

3.2.2 Feature subset selection

An exhaustive search of all possible subsets is usually computationally too expensive. Even for a relatively small number of features, p, the number of possible subsets is $2^p - 1$. In practice, it is very difficult to perform an exhaustive search, and other less computationally expensive techniques have been identified. Spolaôr *et al*. (2013) distinguish between three multi-label FS approaches, namely the wrapper, embedded, and filter approach. *Wrappers* in essence use the learning algorithm to score subsets of the features according to their predictive power. *Embedded* methods are dependent on the learning algorithm and FS is performed during training. For *Filters*, on the other hand, FS is independent of the learning algorithm, and subsets of features are selected during pre-processing. These approaches are now described in more detail.

Wrapper approach

The wrapper approach offers a simple and efficient way to address FS (Kohavi and John, 1997). Wrappers require a specific learning algorithm to evaluate and determine which features are to be selected. Applying the wrapper approach finds features that are better suited for the specific learning algorithm but comes at a higher computational cost as it calls the learning algorithm for each feature set selected. The learning algorithm is used to score subsets of the features according to their ability to predict.

The process can be summarised as follows:

(i) Search the space for all possible feature subsets.

(ii) Assess the predictive performance of a learning algorithm on each subset.

(iii) Choose a classifier to use.

A schematic overview of the wrapper approach is given in Figure 3.1.

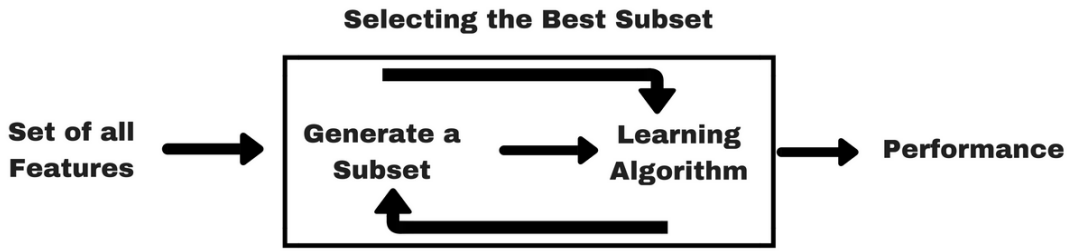


Figure 3.1 Schematic representation of the wrapper approach (Source: Kaushik, 2016).

This approach is feasible if the number of features is not too large, but for a large number of features, it becomes computationally expensive. Kohavi and John (1997) provide a review of a wide range of search strategies that can be employed.

Forward selection and backward elimination are examples of wrapper methods which are commonly used. Forward selection is an iterative approach which starts with no features in the model. At each iteration, the feature which most improves the model is added. This is repeated until the addition of a new feature does not improve the performance of the model. In backward elimination, the initial model includes all the features and the least significant feature is removed at each iteration. The process is repeated until no further improvement in the performance of the model is observed on the removal of features. Wrappers have been criticised in the literature for being a brute force method that requires a large amount of computational power (Kashef *et al*., 2018).

Embedded approach

Some learning algorithms are developed to include FS as part of the training step in the procedure (for example, decision trees). The approach is embedded in the learning algorithm and, in the case of classification problems, decides at each stage which feature is able to best distinguish between classes. These embedded methods are specific to the selected learning

algorithm. Regularisation methods are often used in this setting. Regularisation methods introduce additional limitations into the optimisation of an algorithm that prejudices the model towards lower complexity, *i.e.* fewer coefficients. Popular examples of regularisation algorithms are least absolute shrinkage and selection operator (lasso), elastic net, and ridge regression. A schematic representation of the embedded approach is given in Figure 3.2.

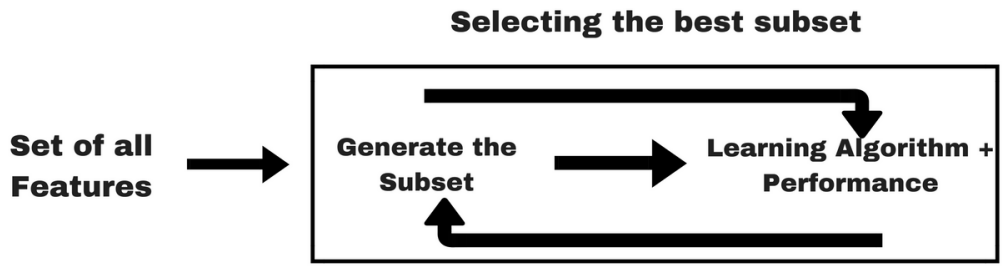


Figure 3.2 Schematic representation of the embedded approach (Source: Kaushik, 2016).

Guyon and Elisseeff (2003) argue that embedded methods could be more efficient than the wrapper methods due to the following two reasons:

1. The more efficient use of data due to the fact that the training data does not need to be split into a training and a validation set; and
2. a solution is reached faster due to the fact that predictors are not retained from scratch for every feature set that is investigated.

Filter approach

The filter approach is independent of the learning algorithm. This implies that, unlike wrappers, the filter approach may not choose the best features for specific learning algorithms. Filters essentially rank the features according to a scoring criterion and then select all the features above some threshold. The filter and embedded approaches return either a subset of features, or weights for all p features. The weights provide a measure of feature importance for all features (not only the features to be included – a threshold still needs to be determined). A schematic representation of the filter approach is given in Figure 3.3.

Figure 3.3 Schematic representation of the filter approach (Source: Kaushik, 2016).

The filter techniques aim to remove irrelevant features by considering the general characteristics of the data. A general procedure for these methods in a multi-label setting can be summarised as follows:

1. Transform the multi-label dataset into q sets of single-label datasets using BR.
2. Quantify the relevance of the first feature for each label. Popular measures in this regard include ReliefF and Information Gain. These measures will be discussed in Section 3.3.2.
3. Repeat step (2) for every feature in order to obtain a score for each feature with regard to each label.
4. Calculate an averaged measure for each feature. This single averaged value provides a global relevance measure for each feature.
5. FS is then performed by ranking the features according to their global relevance measures and only selecting the features that correspond to the measures that exceed a specified threshold value. The selection of an appropriate threshold is often not straightforward. Dreyfus and Guyon (2006) recommend that random probes are generated to determine the appropriate threshold to use. More information on probe variables will be provided in Section 3.3.5.

The filter approach has the advantage that it is fast and easy to implement. This is due to the fact that only p scores or measures need to be calculated – one for each feature. The major limitation of filters for FS is that the approach does not enable the user to identify redundant features. Features which are redundant are likely to have similar rankings. Spolaôr *et al.* (2012a) note that filters are used more frequently in research on multi-label FS than the other two approaches.

A large amount of research has been focussed on FS for the single-label classification problem. However, limited results have been presented for the multi-label case. In the following section, background on FS in the multi-label context will be provided.

3.3 Multi-label feature selection

In this section, the focus will be on FS when multiple labels can be associated with each instance. Due to the complexity of multi-label classification problems, the process of FS is more difficult than in the case of the single-label problem. This complexity is due to the fact

that the FS problem is not limited to the interactions between the features and a single label, but that there are multiple labels to consider, and in addition, there could be interactions among these multiple labels.

This section will be divided as follows. In Section 3.3.1 the importance of FS in the multi-label context will be examined. In the section thereafter, a discussion of the relevance measures that are available in a filter approach will be presented. In particular, this section will focus on the manner in which the three relevance measures used in this study are applied in the empirical study of Chapters 4 and 5. The three relevance measures which are included are Information Gain (IG), ReliefF, and the correlation coefficient.

As is the case with multi-label classification, multi-label FS techniques can be divided into the same two broad categories. Sections 3.3.3 and 3.3.4 will focus on the FS procedures based on problem transformation and so-called "true" multi-label approaches which have been presented in the literature. In Section 3.3.5 the technique presented by Sandrock and Steel (2016) which utilises probe variables is discussed. Finally, in the last section, the method proposed by Spolaôr *et al*. (2013) is presented. The techniques proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013) receive special attention due to the central role that they play in the analysis performed in Chapter 5.

3.3.1 The importance of feature selection in the multi-label context

FS in the single-label context has received substantial attention in the literature, but due to the relatively recent developments in multi-label data not much has been reported in the literature (Spolaôr *et al.*, 2012a). The performance of a classification algorithm is influenced by the quality of the training data. In a similar manner as in the single-label case, irrelevant and redundant features could decrease the accuracy of a classifier in the multi-label case. It could also limit the speed of the classification algorithm.

Spolaôr *et al*. (2012b), Spolaôr *et al*. (2015), Spolaôr *et al*. (2016), Kashef *et al*. (2018), and Pereira *et al*. (2018) all perform systematic reviews related to multi-label FS. These reviews provide valuable insight into the research that has been conducted in the field of multi-label FS. The authors note that multi-label FS techniques can be divided into approaches that apply some sort of transformation, namely the problem transformation approach, and those techniques that directly address the multi-label data.

Prior to a discussion of these different approaches, it is helpful to revisit the notion of relevance in the multi-label context.

3.3.2 Relevance measures

The importance of features can be determined based on the characteristics of the dataset. As mentioned before, a *relevant* feature is defined to be a feature that is relevant for at least one of the q labels. *Irrelevant* features are features that do not influence any of the multiple labels which are present. *Redundant* features, on the other hand, are features that contain information that has already been included in one or more of the other features, *i.e.* the redundant feature does not contribute any new and/or unique information to improve classification.

In the multi-label setting, it is also useful to distinguish between features that are locally relevant and those features that are globally relevant (Sandrock and Steel, 2016). *Local* relevance refers to the relationship between a predictor and a single label. A locally relevant feature is relevant for a given label if it explains the label, irrespective of its relevance to any other labels. On the other hand, the *global* relevance of a feature is an indicator of the relationship between a single feature and all the labels. A feature is said to be globally relevant for all labels it if can explain all the labels effectively.

This discussion is made easier if some notation is introduced.

For feature X_i, $i = 1,...,p$ and label Y_j, $j = 1,...,q$ let $\mathbf{A}: p \times q = \left[A_{ij} \right]$ be a matrix with entries

$$A_{ij} = \begin{cases} 0 & \text{if feature } i \text{ is deemed irrelevant for label } j \\ 1 & \text{if feature } i \text{ is deemed relevant for label } j \end{cases}.$$

The entries in this matrix are dependent on the relevance measure used to decide whether X_i is relevant for label j.

The most popular relevance measures are Information Gain (Chen *et al*., 2007, Wei *et al*., 2009, Dendamrongvit *et al*., 2011, Lastra *et al*., 2011, Spolaôr *et al*., 2012a, Lee and Kim, 2013, Spolaôr *et al*., 2013, Cai *et al*., 2015), ReliefF (Spolaôr *et al*., 2011, Spolaôr *et al*., 2012a, Kong *et al*., 2012, Spolaôr *et al*., 2013, Cai *et al*., 2015), and correlations (Gu *et al*., 2011, Braytee *et al*., 2017, Weng *et al*., 2018, Han *et al*., 2019). Other relevance measures used less frequently are the Fisher Score, Chi-square, Gini index (Trohidis *et al*., 2008, Zhao *et al*., 2011), mutual

information (Doquire and Verleysen, 2013), category contribution (Zhang and Duan, 2019) and rough set (Slezak and Ziarko, 2003).

A second important aspect that influences the entries in the relevance matrix is the threshold which is applied to determine relevance. For the relevance matrix, features corresponding to entries above a specified threshold are deemed to be relevant. The procedure used to determine the appropriate thresholds will be discussed in Sections 4.4.1 and 5.3.1.

A measure of global relevance may be computed based on the individual local relevance scores. The row totals of the matrix *A*, namely $A_{i+} = \sum_{j=1}^{q} A_{ij}$, $i = 1, 2, ..., p$ provide useful information regarding the overall relevance of the features, and a naïve approach would declare feature i globally relevant if $A_{i+} > 0$. Melo and Paulheim (2019) note that the majority of research reported refers to global relevance measures. When performing multi-label FS, it is important that cognisance is taken of this distinction.

In this dissertation, three measures of the strength of the relationship between a given feature and a given label are calculated to determine relevance. These are the correlation coefficient, ReliefF, and IG. A discussion of these relevance measures will be presented next.

Correlation coefficient

Correlation-based FS methods are popular due to their intuitive nature. Irrelevant features will have low absolute correlation with the class, while redundant features will be highly correlated with one or more of the other features (Bolón-Canedo *et al*., 2012).

The absolute correlation coefficients are calculated for each feature X_i, $i = 1, ..., p$ and label Y_j, $j = 1, ..., q$. This leads to a $p \times q$ matrix, $\mathbf{F}$, where F_{ij} are the feature importance scores that quantify the relevance of feature i for label j. In this scenario, the entries of $\mathbf{F}$ are the absolute correlations. Once the absolute correlation coefficients have been calculated, some threshold needs to be applied in order to obtain the relevance matrix, $\mathbf{A}$. For example, consider the scenario where the absolute correlation between the feature 3 and label 4 is equal to 0.46. This means that F_{34} in matrix $\mathbf{F}$ will be 0.46. If the threshold is set to, for example 0.3, this implies that A_{34} in the relevance matrix A will be assigned a value of 1. The selection of an appropriate threshold is not straight-forward. In this dissertation, the choice of threshold will

be driven by the data and ultimately on the multi-label evaluation measures. The procedure followed to determine the appropriate threshold for the benchmark data and the synthetic data will be discussed in Sections 4.4.1 and 5.3.1, respectively.

ReliefF

ReliefF is an extension of the original Relief measure which measures the quality of features. The paper presented by Kira and Rendell (1992) and Robnik-Sikonja and Kononenko (2003) serve as good references on Relief. In a binary classification context, Relief randomly samples an instance, R_i from the data and then finds its nearest neighbour from the same class (called nearest hit, H) and the opposite class (called nearest miss, M). The quality estimation is updated for all features depending on their value for R_i, M, and H. If instance R_i and H have different values of the feature, then the feature separates two instances with the same label which is not desirable so the value w decreases, the process is repeated m times. The number of Relief iterations, m, is user specified. Robnik-Sikonja and Kononenko (2003) provide the following pseudo code for the basic Relief algorithm:

Algorithm Relief

Input: for each training instances a vector of feature values and the label value.
Output: the vector w of estimations of the qualities of features
set all weights $w := 0.0$;
for *number of repetitions* := 1 **to** m **do begin**
    randomly select an instance R_i;
    find nearest hit H and nearest miss M;
    for *feature* := 1 **to** p **do**
    $w := w - diff(feature, R_i, H)/m + diff(feature, R_i, M)/m$:
end;

The function *diff*(*feature*, I_1, I_2) calculates the differences between the values of the feature for the two instances, I_1 and I_2. For discrete features the difference is 1 (if the values are different) or 0 (if the values are equal). For continuous features, the difference is the actual

difference normalised to the interval [0,1]. The normalisation with n guarantees all weights w to be in the interval [-1,1]. The original Relief can deal with both nominal and numerical features, but it cannot handle incomplete data.

The extension to Relief, ReliefF, is more robust; it adds the ability to deal with multiple classes and is capable of dealing with missing data (Bolón-Canedo *et al*., 2012). Both Relief and ReliefF essentially reward a feature for having different values on a pair of nearest examples from different classes and penalises it for having different values on examples from the same class (Demšar, 2010, and Robnik-Sikonja and Kononenko, 2003). ReliefF also selects a random instance R_i, but then searches for k nearest neighbours from each of the different classes, namely nearest misses and nearest hits. The parameter k is user-defined and Konenenko (1994) show that the default value of ten is appropriate. The update formula is similar to that of Relief, except that the contribution of all the hits and misses are now averaged. This version of ReliefF is implemented as `ReliefFequalK` in the R library `CORElearn`. In this dissertation, the ReliefF algorithm where k nearest neighbours have weight which decreases exponentially as the rank increases is used. It is implemented as `ReliefFexpRank` in `CORElearn`. The rank of the nearest instance is determined by increasing the Manhattan distance from the randomly selected instance, R_i.

ReliefF also outputs a value w – which ranges from –1 to 1 – for each feature. A large positive w is assigned to important features. The value w is calculated for each feature X_i, $i = 1,..., p$ and label Y_j, $j = 1,..., q$. Once again, this leads to a $p \times q$ matrix, $\mathbf{F}$, of w-values. A threshold needs to be applied in order to obtain the relevance matrix, $\mathbf{A}$. For example, consider the scenario where the w-value between the feature 3 and label 4 is equal to 0.82. This means that F_{34} in matrix $\mathbf{F}$ will be 0.82. If the threshold is set to, for example 0.05, this implies that A_{34} in the relevance matrix A will be assigned a value of 1. There are two other parameters for the procedure using ReliefF, namely the number of ReliefF iterations, m, and the significance level, α (Robnik-Sikonja and Savicky, 2018). A threshold value, τ, is calculated by $1/\sqrt{\alpha m}$. For a fixed number of ReliefF iterations, the threshold, τ, will therefore increase as the significant level decreases. If the significance level is fixed, τ will increase as the number of ReliefF iterations decreases. In both of these cases, this will lead to fewer features that are deemed to be relevant. More information on the choices of α and m will be provided in Sections 4.4.1 and 5.3.1.

The main advantage of ReliefF compared to measures such as the correlation coefficient is that it takes the effect of interacting features into account as well.

Information Gain

IG is commonly used as a measure of feature relevance in filter strategies that evaluate individual features (Yang and Pedersen, 1997 and Pereira *et al*., 2015). The IG measure is based on the concept of entropy. Let D be a dataset with p attributes, and a single binary label. Pereira *et al*. (2015) define the entropy of the class distribution in D as

$$entropy(D) = -\sum_{i=1}^{2} p_i \log p_i$$

where p_i is the probability that an arbitrary instance in D is associated with the label.

The IG values are calculated on the set of features $X_1, X_2, ..., X_p$ and takes the difference between the entropy of the dataset and the weighted sum of the entropies of the subsets of the data. Larger IG values imply a strong association between the feature and the labels. The IG values can be calculated by

$$IG(D, X_j) = entropy(D) - \sum_{v \in X_j} \frac{|D_v|}{|D|} entropy(D_v)$$

where X_j, $j = 1, ..., p$ represents the features in dataset D, $D_v \subseteq D$ where D_v consists of all the examples where $X_j = v$. Note that the cardinality of the dataset is denoted by $|D|$.

IG is one of the most popular attribute evaluation methods (Bolón-Canedo *et al*., 2012). It has the advantage that it provides an ordered ranking of all the features. The features with IG values greater than or equal to a threshold are selected. Once again, the selection of this threshold is not clear, and ideally this decision will be driven by the characteristics of the data. Bolón-Canedo *et al*. (2012), for example, select all features with a positive IG value.

Spolaôr *et al*. (2015) and Pereira *et al*. (2018) note the need to consider a taxonomy specific for multi-label FS. The paper by Pereira *et al*. (2018) provides such a taxonomy, and the categorisation recommended in this article will be used to structure the discussion in the next section. In Section 3.3.3, a review of the problem transformation approaches will be provided.

3.3.3 Feature selection strategies based on problem transformation approaches

As was the case for multi-label classification, the complexity of multi-label datasets means that it could prove useful to perform some form of data transformation prior to performing FS. This is necessary irrespective of whether one is interested in the local or global relevance of the features. In this section, the problem transformation approaches that relate to FS will be discussed.

When using the problem transformation approach to FS, the multi-label dataset is transformed into several single-label datasets in the same manner as described in Section 2.4. In this chapter, these transformation approaches will be presented in the context of FS.

Once the multi-label dataset has been transformed to a single-label dataset, typically by applying either BR or LP transformations (Spolaôr *et al*., 2012b), a traditional FS technique can then be applied to the transformed data. For example, Chen *et al*. (2007) compare the fundamental *copy*, *select-max*, and *select-min* transformations with a new problem transformation technique based on entropy. After the FS has been applied, the classification algorithm can be run on either the reduced single-label dataset or the reduced multi-label dataset.

BR FS has proven particularly popular and has been employed by a number of researchers. When using a BR transformation, features are selected independently for each binary dataset. The results from the FS steps are then combined in some way, for example, by averaging the results over all q binary datasets. Pereira *et al*. (2018) distinguish between two different approaches to combining the datasets that results from FS using BR, namely the *External* and *Internal* approaches. The External approach combines the FS selection result for each binary model into one output. The reduced dataset is then used as input for a multi-label classifier. There is no need for any aggregation to be performed prior to performing classification. The Internal approach applies the classification algorithm directly to each of the single-label datasets. Once the FS step has been completed, each reduced single-label dataset is used as an input for a single-label classifier. The results are only combined after the classification step has been completed.

Yang and Pedersen (1997) evaluate relevance measures such as IG, mutual information, and the chi-square statistic in a multi-label text categorisation problem. The authors evaluate each label individually which is equivalent to performing a BR transformation. The kNN algorithm

is employed after the FS step. The authors show that up to 98% of the features could be removed when using IG and chi-square relevance measures without losing categorisation accuracy (as measured by average precision). Rogati and Yang (2002), Zheng *et al*. (2004), and Olsson and Oard (2006) also apply BR in the text categorisation setting using IG and chi-square statistic as relevance measures.

BR, in conjunction with ReliefF and IG, is employed for FS by Spolaôr *et al*. (2013). The authors compare the BR transformation with the LP transformation using the same relevance measures. The results show that the methods which utilise ReliefF perform better than those using IG. The authors note that this can be explained by the fact that ReliefF considers the interaction among features. Counterintuitively, they also find that there is very little difference between the performance of BR and LP. One would expect LP to perform better since LP takes the label interactions into account. More detail in this regard will be provided in Section 3.3.6.

Dendamrongvit *et al*. (2011) use BR based on the Internal strategy. The authors discovered that each label in text categorisation is explained by a different set of features. They used a separate FS technique for each label and sent the output to a single-label classifier.

In the following section, FS techniques that deal directly with multi-label data are presented.

3.3.4 Feature selection strategies based on multi-label approaches

"True" multi-label FS methods are adaptations of established FS techniques and they do not require any transformation of the data prior to the FS step. In this section, a brief discussion of some of the "true" multi-label approaches will be presented, along with some comments on the performance of these methods.

Multi-label Latent Semantic Indexing (MLSI)

Some of the multi-label approaches are adapted from single-label approaches, such as Latent Semantic Indexing (LSI) or Principal Component Analysis (PCA). These techniques reduce the number of features by removing features which are deemed to be irrelevant or by creating a projection of the feature space. Once the number of features has been reduced, the procedure provides a ranking of the features which are retained.

Yu *et al*. (2005) propose a multi-label extension of LSI, called MLSI, which is based on mutual information. LSI uses Singular Value Decomposition (SVD) to identify patterns between terms and concepts in unstructured text data. The technique is based on the notion that words that

are used in the same contexts tend to have meanings that are similar. LSI is unable to incorporate additional information, for example, if a document is labelled, useful information about the content would be reflected in these labels. In multi-label text settings, a document can typically be assigned to multiple categories simultaneously. MLSI extends LSI by retaining the information of the input features and incorporating the correlations between multiple outputs. It is a feature extraction technique based on dimensionality reduction.

Multi-label Dimensionality Reduction via Dependence Maximisation (MDDM)

Zhang and Zhou (2010) propose a method that is an adaptation of dimensionality reduction using PCA aimed at solving multi-label problems. The proposed technique produces a ranking of features by maximising the dependence between the original feature description and the corresponding class labels. The authors compare MDDM with methods such as PCA and MLSI using the ML-kNN classifier on eleven multi-label datasets that are based on Yahoo webpages. MDDM performs well when compared to MLSI.

Multi-label ReliefF

Due to the effectiveness of the ReliefF measure, several adaptations of the algorithm for multi-label data have been proposed by Read (2008), Pupo *et al*. (2013), Spolaôr *et al*. (2013), Slavkov *et al*. (2013), Spolaôr and Monard (2014), and Spolaôr *et al*. (2015). Many of the multi-label ReliefF extensions are based on problem transformation methods, where the final weights for each feature are calculated using some aggregation strategy. Popular aggregation strategies include the average, minimum, and maximum (Reyes *et al*., 2013).

Reyes *et al*. (2013) propose three multi-label extensions to FS using ReliefF, namely ReliefF-ML, PPT-ReliefF, and RReliefF-ML. ReliefF-ML can be seen as a generalisation of the classic ReliefF algorithm with a modified equation for the updating of the weights. PPT-ReliefF uses the Pruned Problem Transformation method (PPT) proposed by Read (2008) which transforms the original multi-label dataset into a new multi-class dataset. The third technique, RReliefF-ML applies the adaptation of ReliefF to regression problems proposed by Robnik-Sikonja and Kononenko (1997). The results of this study show that the three proposed ReliefF extensions performed better than the full model.

The RF-ML procedure proposed by Spolaôr *et al*. (2015) is similar to RReliefF proposed by Reyes *et al*. (2013) and Reyes *et al*. (2015). The two main differences are: a) RF-ML introduces

a dissimilarity function which is able to consider multiple labels at the same time, and b) RF-ML searches for k nearest neighbours. RF-ML does not require any prior transformation which leads to faster implementation. The authors argue that the use of the dissimilarity function leads to greater flexibility for multi-label FS. Two recommendations for dissimilarity measures are the Hamming distance and Jaccard dissimilarity. RF-ML is compared to the approaches using ReliefF based on the problem transformations BR and LP. RF-ML did not lead to significant improvements in performance when applied to ten of the benchmark datasets. Multi-label ReliefF will be discussed in more detail in Section 3.3.6.

Multi-label Naïve Bayes

Zhang *et al*. (2009) utilise the wrapper approach to directly address multi-label data. The technique aims to identify the best feature set. The Multi-label Naïve Bayes classifier proposed by Zhang *et al*. (2009) is adapted to incorporate features selection. The proposed approach has two stages. During the first stage irrelevant and/or redundant features are removed using PCA. This reduces the size of the feature pool. A genetic algorithm (What is the genetic algorithm?, n.d.) is then used during the second stage to select a subset of features. This subset of features is selected using a function which incorporates the dependence among the labels. The study had two aims: (1) to determine whether FS improved classification, and (2) to compare the proposed procedure with other algorithms. The results indicate that FS did lead to improved classification performance. The performance of the proposed classifier is also shown to be better than that of ADTBoost.MH, Rank-SVM, BR+Naïve Bayes and Constrained Non-negative Matrix Factorization.

In the following section, FS based on probe selection will be discussed.

3.3.5 Probe selection

Probe Variables

One of the main challenges in FS is to decide how many features should be retained in the reduced model. The discussion presented in this section is based on the outline of the development of probe selection presented by Sandrock (2013) and Sandrock and Steel (2016). Typically, there are two options when determining the number of features to be selected. One could either determine a threshold that is specified, possibly data-dependently, or one could specify the number of features that should be included.

Independent probes are able to assist the user in making this decision (Tuv *et al*., 2008). The concept rests on the notion that one can add a number of randomly generated features – which are independent of the response variable – to the original "true" set of features. A successful FS technique should then be able to rank relevant features higher than these randomly generated probes, and irrelevant features largely lower. This effectively provides a cut-off point that determines which features should be deemed relevant.

Bi *et al*. (2003) suggest that probe features can be generated from, for example, a normal distribution. Tuv *et al*. (2008) caution that this approach is insufficient, since the original feature values may display some distinct structure that needs to be considered. Due to this consideration, Tuv *et al*. (2008) recommend that random imputed values of the original features should be used. This notion is similar to what is done to determine variable importance in random forests. For more detail on this topic, refer to Hastie *et al*. (2009: 593).

Tuv *et al*. (2008) employ random forests to obtain a relative feature ranking by averaging (across all the trees in the forest) how often features are used in determining the splits of the trees. This leads to a relative feature ranking. One reasonably expects that random forests should assign a higher ranking to relevant features than to the probe features. Tuv *et al*. (2009) note that the proposed method obtains what they refer to as "a threshold for importance" that provides a cut-off point for the inclusion of features. In order to obtain statistical significance, Tuv *et al*. (2008) suggest that the process of generating independent probe variables and ranking features should be repeated a number of times. Sandrock and Steel (2016) propose a novel approach to using independent probes to perform FS on multi-label data.

Multi-label Feature Selection using Independent Probes

Sandrock and Steel (2016) note that an FS technique in a multi-label scenario should be able to identify features whether they are locally or globally relevant. In the multi-label setting, one might find that features that predict one of the labels successfully might not necessarily do the same for other labels. In fact, a feature that predicts one of the labels well, may in fact be irrelevant or redundant for one or more of the other labels.

Sandrock and Steel (2016) recommend that a transformation of the $p \times q$ matrix $\mathbf{F}$ of feature importance scores (introduced in Section 3.3.2) to a matrix $\mathbf{A} = \left[A_{ij} \right]$ of indicator values can be performed. $A_{ij} = 1 \Leftrightarrow X_i$ implies that the feature is locally relevant for label Y_j. In order

for this to be attained, it is crucial to stipulate an appropriate threshold for F_{ij}, where F_{ij} are the feature importance scores that quantify the relevance of feature i for label j. Probe variables provide an intuitive procedure for transforming from $\mathbf{F}$ to $\mathbf{A}$.

The authors denote the matrix $\mathbf{F}$ by $\mathbf{F_{XY}}$ and order each row of $\mathbf{F_{XY}}$ in decreasing order. This provides a ranking of the features, $X_1, X_2, ..., X_p$, according to their importance for label j, $j = 1, ..., q$.

Sandrock and Steel (2016) extend the approach proposed by Tuv *et al*. (2008) to multi-label FS. A probe variable S_i for X_i is added by randomly permuting the values in $\mathbf{x}_{(i)}$, $i = 1, ..., p$. Now let $\mathbf{S}: N \times p \equiv \left[\mathbf{s}_{(1)}, ..., \mathbf{s}_{(p)}\right]$ be the matrix obtained by randomly permuting the rows of $\mathbf{X}$, and write $\mathbf{F_{SY}}$ for the $p \times q$ matrix of feature importance values between $\mathbf{S}$ and $\mathbf{Y}$. This is repeated B times, giving B matrices $\mathbf{S}_1, ..., \mathbf{S}_B$ and the corresponding matrices $\mathbf{F_{SY}}(b)$, $b = 1, ..., B$ are then calculated. Now let $F_{SY}^{(b)}(i, j)$ denote the $(i, j)^{th}$ element of $\mathbf{F_{SY}}(b)$. If X_i is locally relevant for label j, this should ideally be reflected in $F_{XY}(i, j)$ being significantly larger than $F_{SY}^{(b)}(i, j)$.

The authors note that a value α, $0 < \alpha < 1$, could be specified under the following conditions: The value α should be small if there is a high cost implication in deeming a locally relevant feature X_i irrelevant for a label q. If the risk of deeming irrelevant features as relevant is smaller, a larger value of α should be specified.

Sandrock and Steel (2016) denote the $F_{SY}^{(b)}(i, j)$ values in increasing ordered by $w_{ij}(b)$, *i.e.* $w_{ij}(1) < w_{ij}(2) < ... < w_{ij}(\mathrm{B})$. Once α has been set, a value for judging the relevance of X_i for label j is calculated from $c_{ij} = w_{ij}(\lceil \alpha B \rceil)$, where $\lceil x \rceil$ = largest integer $\leq x$. If $F_{XY}(i, j) < c_{ij}$, one can conclude that X_i is not relevant for label j. The entries in the matrix $\mathbf{A}$ can now be calculated by $A_{ij} = Ind(F_{XY}(i, j) > c_{ij})$.

When considering global relevance, one must consider the row totals $A_{+j} = \sum_{i=1}^{q} A_{ij}$. The decision to deem a feature globally relevant will require a threshold for the row totals.

Sandrock and Steel (2016) refer to this threshold as a *label-cut*. The label-cut is defined as the minimum number of labels for which a feature should be deemed locally relevant in order to be deemed globally relevant. If the row total $A_{+j} = q$, feature i is deemed relevant for all the labels. If $A_{+j} = 0$, feature i is irrelevant. The label-cut value can be specified upfront, or it can be determined from the data, for example by using cross-validation. Due to the intuitive interpretability of the label-cut value, it would be useful to specify a single value upfront. The user can decide that all features that are relevant for at least half of the labels will be included. Sandrock and Steel (2016) also note that a stepwise approach could be followed: In the first step all features (if any) are identified where $A_{+j} = q$. In the following step, the features with $A_{+j} = q - 1$ are obtained, and this is continued until the set of features for which $A_{+j} = 0$ is reached. A specified threshold for A_{+j} will determine the final number of features which are deemed to be globally relevant.

Sandrock and Steel (2016) consider three different relevance measures combined with the Probe Selection approach: Probe Selection using the correlation coefficient as relevance measure, Probe Selection using IG as relevance measure, and Probe Selection using ReliefF as relevance measure.

The authors find that the new proposal performs well, and that the output provides useful information when multi-label FS is performed.

In the comparative analysis in Chapter 5, the three Probe Selection techniques proposed by Sandrock and Steel (2016) will be compared to the new method proposed in this dissertation. In the following section, a discussion of the ReliefF based on the BR approach proposed by Spolaôr *et al*. (2011) and Spolaôr *et al*. (2013) will be presented.

3.3.6 ReliefF based on the binary relevance approach

Spolaôr *et al.* (2013) propose and evaluate four multi-label FS techniques using the filter approach where the importance of a feature is evaluated irrespective of any particular classifier. The authors limit their study to the use of the problem transformation methods BR and LP. The relevance measures used are IG and ReliefF. The four methods are:

(i) *RF – BR*: ReliefF based on the BR approach;

(ii) *RF – LP*: ReliefF based on the LP approach;

(iii) *IG – BR*: Information Gain based on the BR approach; and

(iv) *IG – LP*: Information Gain based on the LP approach.

Spolaôr *et al*. (2013) argue that the relevance measures ReliefF and IG enable the search for features that provide a better separation of classes and a reduction in the uncertainty. ReliefF takes the effect of interacting features into account despite evaluating each feature separately.

The *RF-BR* method was originally proposed in Spolaôr *et al*. (2011), but initially the authors only considered a few multi-label datasets. Spolaôr *et al*. (2012a) expand the analysis to more datasets and compare it to the IG method proposed by Clare and King (2001).

RF-BR and *IG-BR* initially transform the multi-label dataset into q binary datasets. Then ReliefF and IG are used to evaluate the set of features $\{X_1,...,X_p\}$ on each of the q binary datasets. These q values for each feature X_i, $i=1,...,p$ are then averaged, and the averages that exceed the specified threshold are the features which are selected. Neither of these methods based on BR consider label correlation. However, Spolaôr *et al*. (2013) note that the methods *RF-LP* and *IG-LP* use the feature importance measure directly calculated from the multi-class dataset which is generated using LP. Therefore, these two methods are able to incorporate the dependencies among the labels.

The empirical evaluation is performed using ten benchmark multi-label datasets from the MULAN repository. The authors use the specific versions of BR and LP, and BRkNN-b which are available on MULAN. The BRkNN-b algorithm is executed with $k=5$. The threshold for both ReliefF and IG was set to 0.01. The authors consider this to be a conservative threshold.

Spolaôr *et al*. (2013) show that the methods that use ReliefF as a feature evaluation measure more often select a smaller number of features than the ones that use IG, with no degradation of the correspondent classifiers. This could be due to the fact that ReliefF considers interactions among features.

In this study, the focus is on the BR problem transformation. For this reason, the *RF-BR* approach by Spolaôr *et al*. (2013) will be used in the comparative analysis performed in Chapter 5.

3.4 A new method based on the methodology of MCA biplots

The main contribution made in this study, namely a new method for multi-label FS that utilises the methodology of MCA biplots, is described in this section. It is important to keep in mind that this dissertation does not aim to expand or contribute to the body of research on MCA biplots. Instead, the goal is to use some aspects of MCA biplot methodology to make a contribution to the field of feature selection in the multi-label context. With this in mind, a brief introduction to biplots, CA, and MCA will be provided in Sections 3.4.1 to 3.4.3. This is followed by a detailed discussion of the new proposal for FS in a multi-label setting.

3.4.1 Biplots

Biplots provide a graphical representation of observations and features simultaneously in two or more dimensions. Biplots were first introduced by Gabriel (1971) and further developed by Bradu and Gabriel (1978), Gabriel and Zamir (1979), Gabriel (1981), and more recently, Gower and Harding (1988), Gower (1990, 1992), and an authoritative text on biplots by Gower and Hand (1996). The aim of biplots (Cox and Cox, 2000) is to find a space in which points representing objects are plotted, and upon which a "framework" is overlaid representing the variables.

The first biplots that were developed were based on PCA – the objects are represented by points in a sub-space of the original space spanned by the features of the data matrix and the original features are represented by vectors plotted in this subspace. The classic biplot is a representation of the rows and columns of a matrix as vectors in a two-dimensional space. The "bi" in biplot refers to the fact that two sets of points (*i.e.* the rows and columns of the target matrix) are visualised by scalar products, not the fact that the display is usually two-dimensional. The biplot and its geometry hold for spaces of any dimensionality (Greenacre, 2010), but one requires dimension-reducing techniques when data matrices have high dimensionality and a graphical representation with two or three dimensions is required.

3.4.2 Correspondence Analysis biplots

CA is a versatile method for visualising data in a space of low dimension, based on the Singular Value Decomposition (SVD) of a matrix. It is primarily used to visualise contingency tables that are constructed when it is possible to place events into two or more sets of categories. A brief historical account of the development of CA is given by Greenacre (1984).

As in PCA, CA aims to reduce the dimensionality of a data matrix and to visualise it in a low-dimensional subspace. It is a statistical visualisation technique that provides a graphical representation of the associations between the levels of a two-way contingency table, but it can also be extended to frequency tables, ratio-scale data, binary data, preferences, and fuzzy-coded continuous data (Greenacre, 2010).

The observed association is represented by the cell frequencies, and one needs to determine whether certain levels of one characteristic are associated with some levels of another. CA biplots display the rows and columns of the data matrix in a low-dimensional space in such a manner that the positions of the row and column points are consistent with their associations in the table. For more information on CA, refer to Greenacre (2007).

3.4.3 Multiple Correspondence Analysis

MCA is an extension of simple CA of two categorical variables to the case of several categorical variables (Greenacre, 2010: 89). MCA differs from PCA in the sense that instead of the $n \times p$ data matrix $\mathbf{X}$ in PCA, there is an $n \times p$ matrix with columns providing the category levels of the p categorical variables for each of the n samples. The discussion on MCA in this section follows the discussion and example outlined by Gower *et al*. (2011). In this regard, Table 3.1 represents a small dataset that contains a categorical variable *Hair Colour* that has four category levels *Dark*, *Grey*, *Fair*, and *Brown*. Table 3.1 provides an example of a data matrix of categorical variables.

Table 3.1: A data matrix for information on five categorical variables for seven individuals.

Case	Sex	Hair Colour	Region	Work	Education
George (1)	M	Brown	England	Manual	School
Alisdair (2)	M	Dark	Scotland	Clerical	University
Jane (3)	F	Brown	Scotland	Professional	University
Ivor (4)	M	Grey	Wales	Professional	University
Myfanwy (5)	F	Fair	Wales	Clerical	School
Harriet (6)	F	Brown	England	Manual	School
Jeremy (7)	M	Grey	England	Professional	Postgrad

Source: Gower *et al*. (2011)

Nominal data such as this is typically coded into numerical proxy variables where, for example, the k th variable is logged in a $n \times L_k$ matrix $\mathbf{G}_k$. L_k denotes the number of category levels that are present for the k th variable, for *Hair Colour* $L_k = 4$. The i th row of the matrix $\mathbf{G}_k$ contains a single unit in the column which corresponds to the category level taken by the i th sample. The other entries in the i th row are all zeroes.

The sums of the columns of $\mathbf{G}_k$ are the frequencies of each category level in the n samples. These frequencies are denoted by $\mathbf{L}_k$, which will be considered as a diagonal matrix. Therefore, $\mathbf{G}_k\mathbf{1} = \mathbf{1}$ and $\mathbf{1}'\mathbf{G}_k = \mathbf{1}'\mathbf{L}_k$; also $\mathbf{1}'\mathbf{L}_k\mathbf{1} = n$ as every sample must contain one category level. $\mathbf{G}_k$ is referred to as an indicator matrix.

Consider the example above: recoding Table 3.1 as an indicator matrix $\mathbf{G}$, where $\mathbf{G}_1$ has two category levels (*Male* and *Female*), $\mathbf{G}_2$ has four category levels (*Brown*, *Dark*, *Grey*, and *Fair*), $\mathbf{G}_3$ has three category levels (*England*, *Scotland*, and *Wales*), $\mathbf{G}_4$ has three category levels (*Manual*, *Clerical*, and *Professional*), and $\mathbf{G}_5$ has three category levels (*School*, *University*, and *Postgrad*), Table 3.2 is obtained.

Table 3.2: Recoding of Table 3.1 as an indicator matrix.

Case	Sex		Hair colour				Region			Work			Education		
	M	**F**	**B**	**D**	**F**	**G**	**E**	**S**	**W**	**M**	**C**	**P**	**S**	**U**	**P**
George (1)	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0
Alisdair (2)	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0
Jane (3)	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0
Ivor (4)	1	0	0	0	0	1	0	0	1	0	0	1	0	1	0
Myfanwy (5)	0	1	0	0	1	0	0	0	1	0	1	0	1	0	0
Harriet (6)	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0
Jeremy (7)	1	0	0	0	0	1	1	0	0	0	0	1	0	0	1
Frequencies	4	3	3	1	1	2	3	2	2	2	2	3	3	3	1

Source: Gower *et al*. (2011)

The indicator matrix $\mathbf{G}$ for the entire dataset is a combination of all the indicator sub-matrices:

$$\mathbf{G} = \left[\mathbf{G}_1 \quad \mathbf{G}_2 \quad \mathbf{G}_3 \quad \ldots \quad \mathbf{G}_p\right] : n \times L$$

where $L = L_1 + L_2 + ... + L_p$. The frequencies $\mathbf{1}'\mathbf{L}_1$, $\mathbf{1}'\mathbf{L}_2$, $\mathbf{1}'\mathbf{L}_3$, $\mathbf{1}'\mathbf{L}_4$, and $\mathbf{1}'\mathbf{L}_5$ are given in the final row of Table 3.2. Gower *et al*. (2011) note that the column sums provide the frequencies of all the levels assumed to be held in an $L \times L$ diagonal matrix $\mathbf{L} = \text{diag}(\text{diag}(\mathbf{L}_1), \text{diag}(\mathbf{L}_2), \ldots, \text{diag}(\mathbf{L}_p))$.

Gower *et al.* (2011) suggest that one generalisation of CA is to treat the indicator matrix $\mathbf{G}$ as if it were a two-way contingency table. This is similar to the CA of chi-squared distance "where the two-way contingency table is treated as if it were a data matrix where the rows or the columns are treated as if they were variables".

The CA of a contingency table $\mathbf{X}$ requires the construction of $\mathbf{R}^{-\frac{1}{2}}\mathbf{X}\mathbf{C}^{-\frac{1}{2}}$, where the matrices $\mathbf{R}$ and $\mathbf{C}$ are the diagonal matrices of row and column masses, respectively (Greenacre, 2007: 201 – 202). When performing an analysis on $\mathbf{X}$, one has $\mathbf{R} = p\mathbf{I}$ and $\mathbf{C} = \mathbf{L}$, that leads to

$$p^{-\frac{1}{2}}\mathbf{G}\mathbf{L}^{-\frac{1}{2}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}',$$

where $\mathbf{\Sigma}$ is the diagonal matrix of positive singular values in descending order, and $\mathbf{U}$ and $\mathbf{V}$ refer to the second and subsequent columns of the SVD. The factor $p^{-1/2}$ is unnecessary but is included to maintain the connection to CA. As in CA, the first singular vectors, corresponding to a unit singular value, can be disregarded, since it is equivalent to working in deviations from the column means. If an SVD is performed on a non-centred matrix, the first (largest) singular value will always be equal to one. Refer to Cox and Cox (2000: 188) for more information on this topic. These vectors are $\mathbf{1}$ and $\mathbf{L}^{1/2}\mathbf{1}$ which, through the orthogonality properties of singular vectors, imply that the remaining singular vectors satisfy

$$\mathbf{1}'\mathbf{U} = \mathbf{0}' \text{ and } \mathbf{1}'\mathbf{L}^{\frac{1}{2}}\mathbf{V} = \mathbf{0}'.$$

As for simple CA, the standard choice for MCA is to use chi-squared distance. For approximating the row chi-squared distances,

$$\mathbf{Z}_0 = \mathbf{U}\mathbf{\Sigma}$$

plotted. A distance measure is preferable to, for example, the Pearson residuals, since the aim is to optimally utilise the distances between the rows. Since the Euclidean distance is only appropriate for continuous data, the weighted Euclidian distance, namely chi-squared distance, is used. The matrix $\mathbf{Z}_0$ is a $p \times 2$ matrix of coordinates. This graphical representation of $\mathbf{Z}_0$ provides a visual representation of the row chi-square differences. The columns are represented by the projected category-level points,

$$\mathbf{Z} = p^{-\frac{1}{2}}\mathbf{L}^{-\frac{1}{2}}\mathbf{V}.$$

These representations allow for the comparison of the categorical variables based on their similarity. When considering the problem of FS in the multi-label setting, the following question was posed in Chapter 1: How does one go about grouping the features? Could a biplot based on performing MCA on the relevance matrix allow one to group the features based on their similarity? In this dissertation, it is proposed that an MCA is performed on the relevance matrix to identify which features are irrelevant (*i.e.*, which features are not considered relevant for any of the labels) and which features can be grouped together since they provide similar information. It should be noted that MCA can be performed on either the indicator matrix or the Burt matrix. The Burt matrix is the matrix of all two-way cross-

tabulations of the categorical variables. In this study, the indicator matrix was chosen. The rationale behind this choice is as follows: If one analyses the indicator matrix in the multi-label FS context, it allows for the direct representation of the features as points in a geometric space. For more information on the Burt matrix, see Gower *et al*. (2011: 372).

The R libraries `UBbipl` and `UBFigs` (Le Roux and Lubbe, 2013) are used to perform the MCA and to construct an MCA biplot that provides a graphical representation of the different groups of features (feature groups). The MCA biplots are independent of the classifier and the biplots obtained from the MCA are used to construct the feature groups used later by both the SVM and XGBoost classifiers.

The features in each feature group are ranked according to either the absolute correlation coefficient, the IG values or the w-values obtained from ReliefF. In the following section, a detailed discussion of the proposed algorithm is presented.

3.4.4 Relevance pattern feature selection

In this section the detailed discussion of the steps for a proposal that utilises MCA biplot methodology to group features with the aim of proposing three different FS procedures are presented. This discussion is followed by a brief summary of the algorithm.

Step 1: Construct the relevance matrix

The first step involves the construction of the relevance matrix $\mathbf{A}$ described in Section 3.3.2. This matrix contains the associations between the p features and q labels. The relevance matrix is obtained by transforming a $p \times q$ matrix, $\mathbf{F}$, where F_{ij} are the feature importance scores that quantify the relevance of feature i for label j. The entries of $\mathbf{F}$ are relevance scores calculated using one of the three relevance measures utilised in this dissertation, namely the correlation coefficient, ReliefF or IG.

Consider the following example for the case where one is interested in determining the association between four features, $p = 4$, and three labels, $q = 3$. For example, in the matrix, $\mathbf{F}$, presented below, the entry $F_{12} = 0.87$ is simply the absolute value of the correlation coefficient calculated between feature 1 and label 2. An absolute correlation coefficient of

0.87 suggests that feature 1 and label 2 are highly correlated and that feature 1 should be considered to be relevant for label 2.

$$\mathbf{F} = \begin{bmatrix} 0.11 & 0.87 & 0.38 \\ 0.06 & 0.03 & 0.31 \\ 0.71 & 0.07 & 0.42 \\ 0.58 & 0.06 & 0.18 \end{bmatrix}$$

Some threshold value is required to construct the relevance matrix, $\mathbf{A}$, from matrix, $\mathbf{F}$. The threshold values for each of these relevance measures are determined empirically based on the characteristics of the data. Threshold values from a predetermined interval of values are selected.

For example, for the correlation coefficient, an interval of threshold values between 0.1 and 0.6 are considered. The SVM classifier is applied to the training and test datasets and the ten evaluation measures are then compared to determine the appropriate threshold value. The process of determining the appropriate threshold is described in greater detail in Sections 4.4.1 and 5.3.1. These same thresholds are then applied when fitting the XGBoost classifier for the results to be comparable. Say, for example, that the appropriate threshold is determined to be 0.4. This means that any feature with a correlation absolute correlation of > 0.4 will be deemed relevant for the particular label. The relevance matrix based on $\mathbf{F}$ is therefore

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

From $\mathbf{A}$, one can now conclude that feature 1 is only relevant for label 2, feature 2 is irrelevant, feature 3 is relevant for labels 1 and 3, and feature 4 is relevant for label 1.

Step 2: Perform MCA on the relevance matrix

Once the relevance matrix is constructed, an MCA is performed on the relevance matrix in order to determine the matrix $\mathbf{Z}_0$. Take note that the relevance matrix, $\mathbf{A}$, is used in the same manner as the data matrix is above. This means that $\mathbf{A}$ is used as the input into the function `MCAbipl()` available in the `UBbipl` library. This matrix allows one to identify features as irrelevant (*i.e.* those features with a row total of zero in the relevance matrix) and also which

features can be grouped together because they provide similar information (redundant features). Greenacre (2010, 89) notes that MCA is an extension of CA of a cross-tabulation of two categorical variables to the case of three or more categorical variables (in our case, labels).

One could argue that MCA could be performed directly on the matrix, $\mathbf{F}$, as the probability of identical rows is far smaller than for indicator values. However, this ignores the critical notions of global and local relevance which are central to the performance of FS in the multi-label setting. For this reason, the relevance matrix is used instead.

Step 3: Construct an MCA biplot

The MCA biplot can now be constructed based on the matrix $\mathbf{Z}_0$. The MCA biplot provides a graphical representation that provides insight into the relationship between the features and the labels, as well as a visual illustration of the different groups of features. An example of such an MCA biplot is given in Figure 3.4 for illustration purposes. The row points are represented by the cadet blue circles, and the column points by the colour squares. The column points are coded in such a manner that each label is represented by one colour. The marker 1 – 0 is used to indicate that label 1 is absent and 1 – 1 is used to indicate that label 1 is present.

The dataset shown consists of 20 features that are plotted based on their relationship to the labels. Six distinct feature groups can be identified based on the MCA biplot.

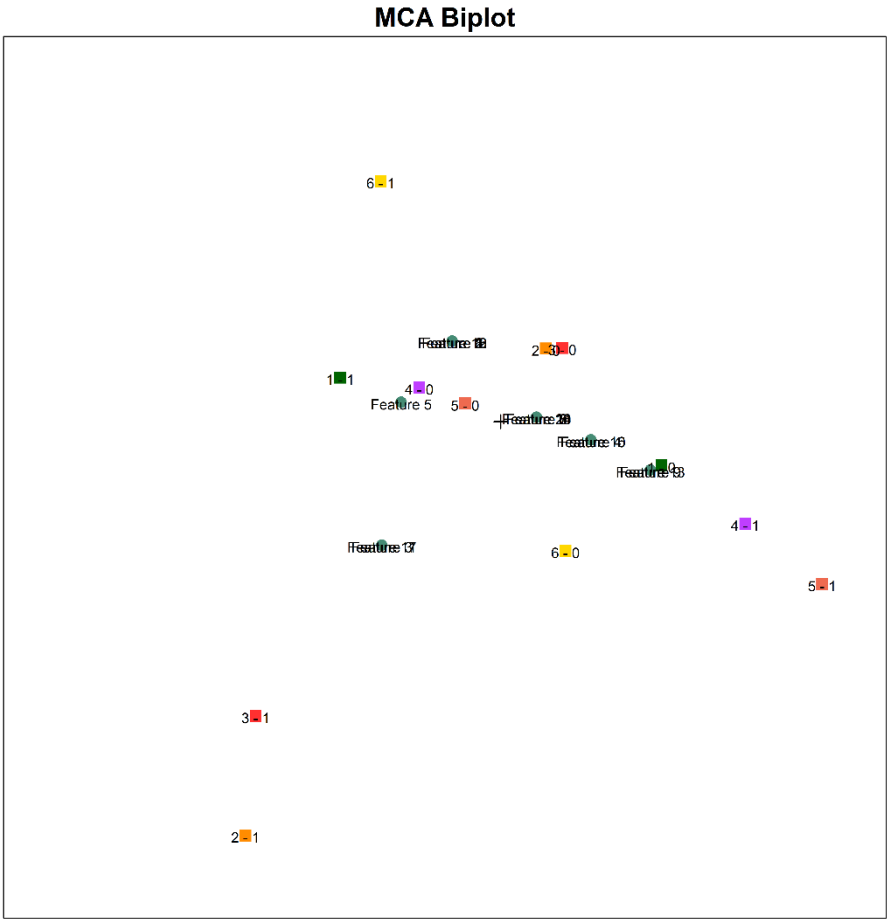


Figure 3.4 An illustrative example of an MCA biplot.

Figure 3.4 also provides information about the labels; however, this is made difficult by the 20 feature markers. The feature markers are removed in Figure 3.5. From this figure one can conclude that the relationship between labels 4 and 5 is stronger than the relationship between labels 4 and 6. This conclusion is based on the column points 4 – 1 and 5 – 1, which plot closer on the MCA biplot than column point 4 – 1 and 6 - 1.

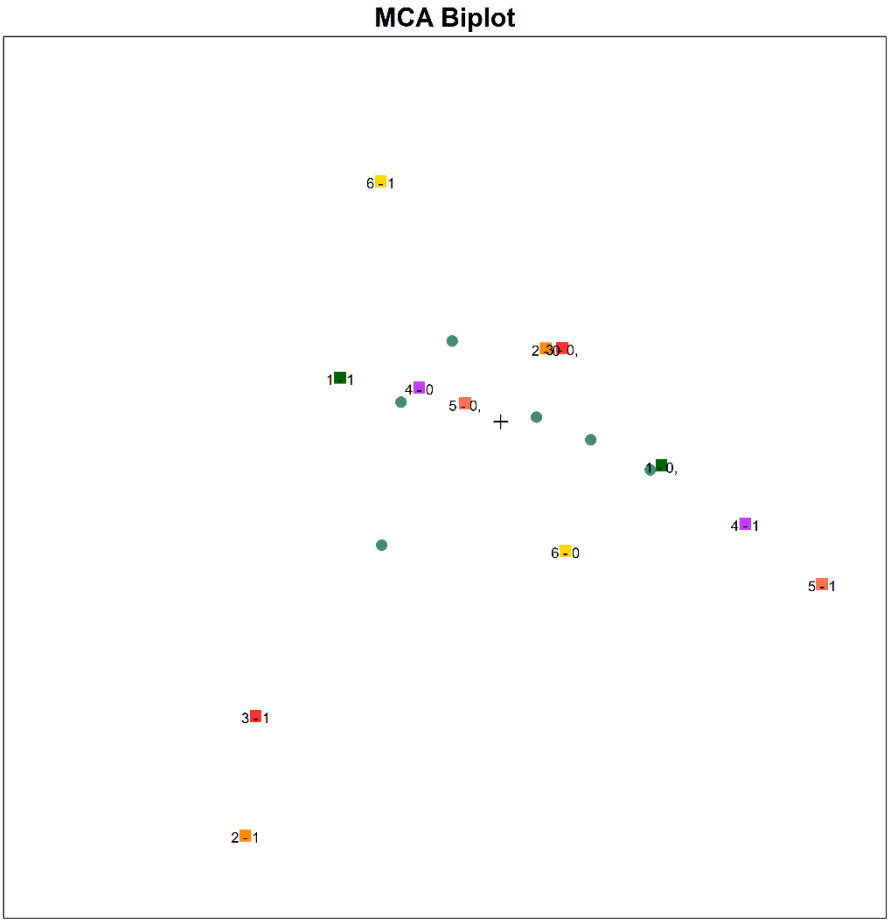


Figure 3.5 An illustrative example of an MCA biplot (without feature markers).

While the column points offer interesting insights, the focus of this dissertation is on feature selection, and for this reason a new plot is introduced which only plots the row points (feature groups). A plot that only contains the row points from the MCA biplot provides a visual representation of the relevance pattern present amongst the features. This plot is no longer a biplot, but a monoplot, since only a single entity (the rows) is represented. The relevance pattern can then be utilised to perform FS, and for this reason the plot is called the relevance pattern feature selection (RPFS) plot in Figure 3.6.

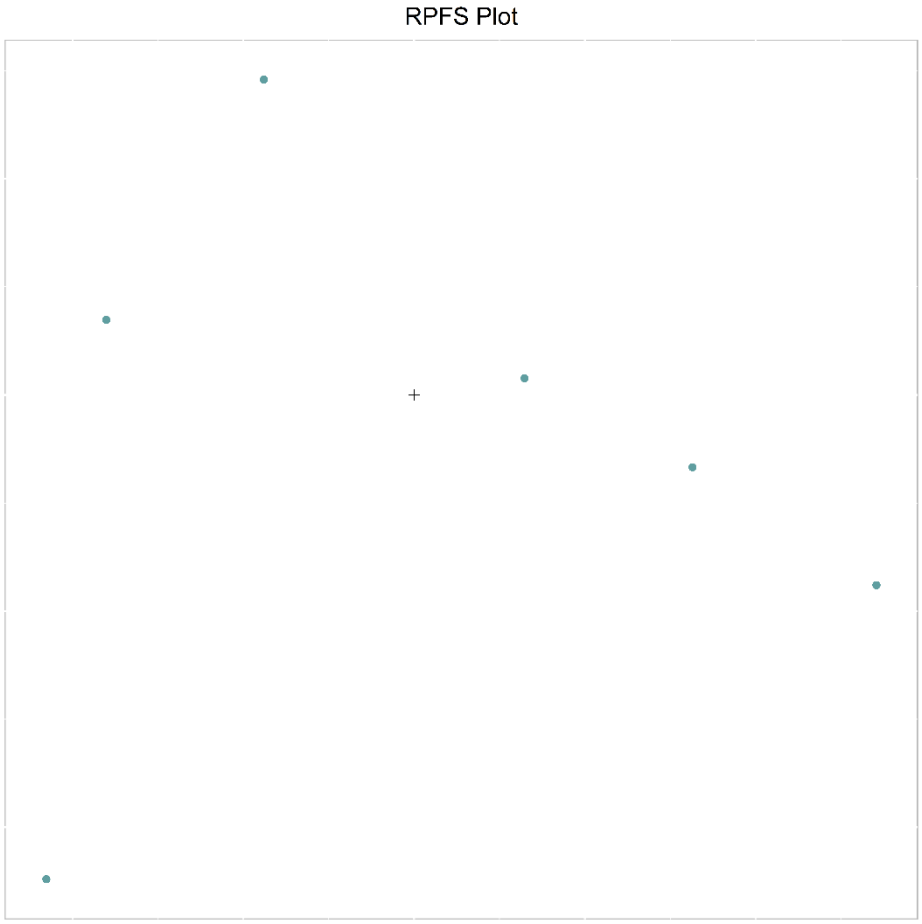


Figure 3.6 An illustrative example of a RPFS plot.

The six distinct feature groups identified from the MCA biplot are clear in the RPFS plot. These feature groups can now be used to perform FS, but first a more detailed discussion of the feature groups is required.

The feature groups can have different sizes, for example, one could have a dataset that is divided into six feature groups as in Table 3.3.

Table 3.3: An illustrative example of the feature groups.

Feature group	Feature
1	1, 2, 8, 12, 16, 19
2	3, 11, 17
3	5
4	9, 13
5	4, 10, 15
6	6, 7, 14, 18, 20

The relevance measures used to determine feature importance provide an inherent ranking of these features within each feature group. Features are ranked according to the values obtained from the relevance measures, *i.e.* the IG values, the absolute correlation coefficients, or the w-values for ReliefF. For illustrative purposes, consider the rankings provided below.

Table 3.4: A ranking of the features in each feature group.

Feature group	Feature
1	19, 1, 8, 16, 12, 2
2	3, 17, 11
3	5
4	13, 9
5	10, 15, 4
6	18, 14, 6, 7, 20

Based on the relevance matrix, one would be able to determine which of these groups contains the features that are deemed to be irrelevant. Assume that the features in feature group 6 are those considered to be irrelevant, for example. Further, if one considers the second feature group, which contains features 3, 17, and 11, one can conclude that these features provide similar information and that at least some of these features are redundant.

The foundation of the proposed approach is to use the groupings of these features to perform FS. The FS approach is based on the notion that features which plot close together provide similar information about the labels.

Step 4: Perform feature selection

Based on the groupings observed in the RPFS biplot and the inherent ranking capabilities of the relevance measures, three different FS procedures are proposed:

Relevant

Remove the group of features identified to be irrelevant. Let k = the number of irrelevant features. Then $\hat{k}$ = the number of features deemed to be irrelevant by the relevance measure. Retain only the $p - \hat{k}$ "relevant" features for the classification. The term "relevant" here refers to the features which are identified to be relevant by the relevance measure. The resulting feature set obtained using this method is provided in Table 3.5, where $\hat{k} = 5$ and $p - \hat{k} = 15$.

Table 3.5: The features included in the model *Relevant.*

Feature group	Feature
1	19, 1, 8, 16, 12, 2
2	3, 17, 11
3	5
4	13, 9
5	10, 15, 4

The resulting feature set no longer includes any features that are deemed to be irrelevant, but a number of redundant features are still present in the feature set. In order to deal with the aspect of redundant features, a number of approaches were considered. These approaches will be discussed next.

Highest

The first option available is to remove all redundant features from the feature set. This implies that one would need to select only one feature from every feature group. Two possibilities were considered. The first involved randomly selecting a single feature from each feature group. This approach was considered initially, but since it ignored the additional information provided by the rankings, it was abandoned for an approach that selected only the highest ranked feature from each feature group. The feature set obtained using this FS approach is presented in Table 3.6.

Table 3.6: The features included in the model *Highest.*

Feature group	Feature
1	19
2	3
3	5
4	13
5	10

One notices that the model Highest leads to a dramatically reduced feature set. Only five features from the original 20 features are included. While this reduction might be attractive in scenarios where there are a fairly large number of feature groups, it might be too restrictive in cases where there are fewer feature groups.

Highest 2

One could remedy this problem by, for example, selecting only the top two or three ranked features from each of the relevant feature groups. This idea could be extended to any number of features and could be included as a user-defined parameter in the procedure. In this dissertation, the decision was made to include the two highest ranked features.

Table 3.7: The features included in the model *Highest 2*.

Feature group	Feature
1	19, 1
2	3, 17
3	5
4	13, 9
5	10, 15

The feature set based on Highest 2 includes nine features, of which four are redundant features. These three reduced sets of features are compared to the full dataset which contains all features.

The proposed RPFS procedure can be summarised as follows:

1) Construct a relevance matrix representing the associations between features and labels using the training dataset.
2) Perform MCA on the relevance matrix.
3) Construct an MCA biplot and RPFS plot.
4) Perform FS, including the following three sets of features:
    a) Select only the features identified as relevant features.
    b) Select the top ranked feature from each of the relevant feature groups.
    c) Select the two top ranked features from each of the relevant feature groups.

In the next section, a short conclusion to the chapter on FS will be given.

3.5 Conclusion

In this chapter, a brief background to the field of FS was presented. Specific mention was made of the different general approaches to FS. In Section 3.3 existing multi-label FS approaches were discussed. The idea of relevance was revisited, and the three relevance measures used in this dissertation were presented and discussed. The technique using Probe Selection presented by Sandrock and Steel (2016), as well as the technique proposed by Spolaôr *et al*. (2013), were discussed in detail. These two techniques are used during the empirical analysis in Chapter 5. Finally, a new method for multi-label FS was presented in Section 3.4. This method utilised the established statistical methodology of MCA biplots to introduce a new

procedure called RPFS to perform FS. In order to aid this discussion, some brief background on the fields of CA and MCA biplots were included.

The new technique described in Section 3.4 will be applied to a benchmark dataset in Chapter 4.

CHAPTER 4

EMPIRICAL INVESTIGATION: BENCHMARK DATASET

4.1 Introduction

In the previous chapter some of the multi-label feature selection (FS) procedures that have been proposed in the literature, were discussed. Specific attention was given to the method proposed in this dissertation: relevance pattern feature selection (RPFS), described in Section 3.4. In this chapter the proposed procedure will be applied to one of the widely used multi-label benchmark datasets. The availability of benchmark datasets makes it possible to objectively compare existing techniques and to evaluate the performance of new methods.

The chapter starts with a short discussion of the benchmark datasets for multi-label classification. These datasets were briefly mentioned in Section 2.2.3 but will be revisited here. Some background information on the *Emotions* dataset is provided in Section 4.3. In Section 4.4, the implementation of the proposed technique is discussed. The results of RPFS applying a support vector machine (SVM) classifier and an extreme gradient boosting (XGBoost) classifier are then presented, followed by a comparison of these two classifiers in Section 4.5. Section 4.6 contains a concise summary of the results.

4.2 Benchmark datasets in multi-label classification

As mentioned in Section 2.2.3 a large part of multi-label research is based on a number of benchmark datasets, which can be found on the MULAN website (Tsoumakas *et al*., 2011). These datasets cover several different domains but are limited with respect to label cardinality and label density. The 26 datasets available on MULAN are summarised in Table 4.1 below. Note that out of the 26 benchmark datasets, 18 are characterised by label cardinalities of less than 3, and 22 are characterised by densities less than 0.1.

Table 4.1: Benchmark datasets from MULAN.

Name	Domain	Instances	Nominal features	Numeric features	Labels	Cardinality	Density
Bibtex	text	7395	1836	0	159	2.402	0.015
birds	audio	645	2	258	19	1.014	0.053
bookmarks	text	87856	2150	0	208	2.028	0.010
CAL500	music	502	0	68	174	26.044	0.150
corel5k	images	5000	499	0	374	3.522	0.009
corel16k	images	13811	500	0	161	2.867	0.018
Delicious	text (web)	16105	500	0	983	19.020	0.019
Emotions	music	593	0	72	6	1.869	0.311
Enron	text	1702	1001	0	53	3.378	0.064
EUR-Lex (1)	text	19348	0	5000	412	1.292	0.003
EUR-Lex (2)	text	19348	0	5000	201	2.213	0.011
EUR-Lex (3)	text	19348	0	5000	3993	5.310	0.001
Flags	images (toy)	194	9	10	7	3.392	0.485
Genbase	biology	662	1186	0	27	1.252	0.046
Mediamill	video	43907	0	120	101	4.376	0.043
Medical	text	978	1449	0	45	1.245	0.028
NUS-WIDE	images	269648	0	128	81	1.869	0.023
rcv1v2 (1)	text	6000	0	47236	101	2.880	0.029
rcv1v2 (2)	text	6000	0	47236	101	2.634	0.026
rcv1v2 (3)	text	6000	0	47236	101	2.614	0.026
rcv1v2 (4)	text	6000	0	47229	101	2.484	0.025
rcv1v2 (5)	text	6000	0	47235	101	2.642	0.026
Scene	image	2407	0	294	6	1.074	0.179
tmc2007	text	28596	49060	0	22	2.158	0.098
yahoo	text	5423	0	32786	31	1.481	0.051
yeast	biology	2417	0	103	14	4.237	0.303

In the next section, the popular *Emotions* benchmark dataset will be discussed in more detail.

4.3 The *Emotions* dataset

4.3.1 Background

Music evokes emotions; it can bring the listener joy, sadness or calm, athletes use it to motivate themselves before sporting events, medical practitioners use it in therapy. The recent growth in digital music libraries poses new and interesting challenges. Most libraries can handle users' queries or searches based on simple classifications such as artist, title or genre. Searches made

based on content similarity have been investigated by Logan and Salomon (2001) and Yang (2001).

Emotion detection is a classic multi-label classification problem, as music signals can be classified into multiple emotion classes simultaneously. More than one label could be associated with an individual signal, *e.g.* a signal could be both *dark* and *mysterious*. The classification of music by emotion is therefore a quintessential multi-label classification problem (Trohidis *et al.*, 2011).

4.3.2 Properties of the *Emotions* dataset

The *Emotions* dataset was constructed as follows: A male subject was asked to listen to sound signals generated from seven different music genres: classical, reggae, rock, pop, hip-hop, and jazz. He was asked to categorise these signals into the ten adjective groups recommended by Farnsworth (1958); he was also asked to add any adjective groups he felt were necessary. The subject added the last three groups (K, L, and M in Table 4.2). Finally, the subject was asked to combine the 13 classes into six super-groups (Li and Ogihara, 2003). The resulting dataset is the well-known emotions from music dataset, which contains 72 music features for 593 songs categorised into one or more of the six super-groups.

Table 4.2: Original thirteen adjective groups.

A	cheerful, gay, happy	H	dramatic, emphatic
B	fanciful, light	I	agitated, exciting
C	delicate, graceful	J	frustrated
D	dreamy, leisurely	K	mysterious, spooky
E	longing, pathetic	L	passionate
F	dark, depressing	M	bluesy
G	sacred, spiritual		

A grouping of these adjective groups leads to the six labels found in the benchmark dataset. The six labels are (A, B), (C, D), (E, L), (H, I, J), (G, K), and (F, M). The final six labels used in the *Emotions* dataset to describe these groupings, are *Happy-Pleased*, *Amazed-Surprised*, *Relaxing-Calm*, *Quiet-Still*, *Sad-Lonely*, and *Angry-Aggressive*. The label cardinality (average number of labels per data case) and label density (the average number of labels, divided by the number of labels) are 1.869 and 0.311 respectively.

The frequency of occurrence of each of the six labels is represented in Figure 4.1. The grouping *Relaxing-Calm* (label 3) occurs most frequently (264 times for the 593 instances) in the benchmark dataset. This is followed by *Angry-Aggressive* (189), *Happy-Pleased* (173), *Sad-Lonely* (168), *Amazed-Surprised* (166), and *Quiet-Still* (148).

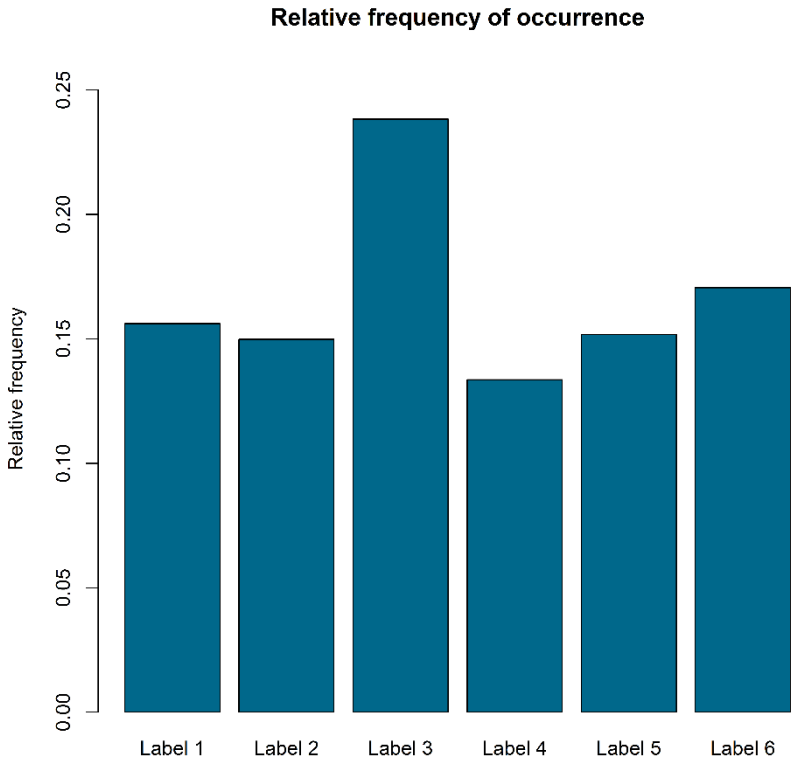


Figure 4.1 Frequency with which each label occurs in the *Emotions* dataset.

When one considers a multi-label dataset, another characteristic of interest is the correlations amongst the labels. With this in mind, the correlation coefficient for each of the label combinations is calculated and shown in a heatmap in Figure 4.2. Darker values represent a higher absolute correlation coefficient. From Figure 4.2, one can note that label 3 and label 6

have the highest absolute correlation of 0.56, while label 4 and label 5 have the second highest correlation coefficient of 0.55. These correlation coefficients are not surprising when one considers the groupings that these labels represent. Label 3 (*Relaxing-Calm*) is negatively correlated with label 6 (*Angry-Aggressive*), and label 4 (*Quiet-Still*) has a positive correlation to label 5 (*Sad-Lonely*).

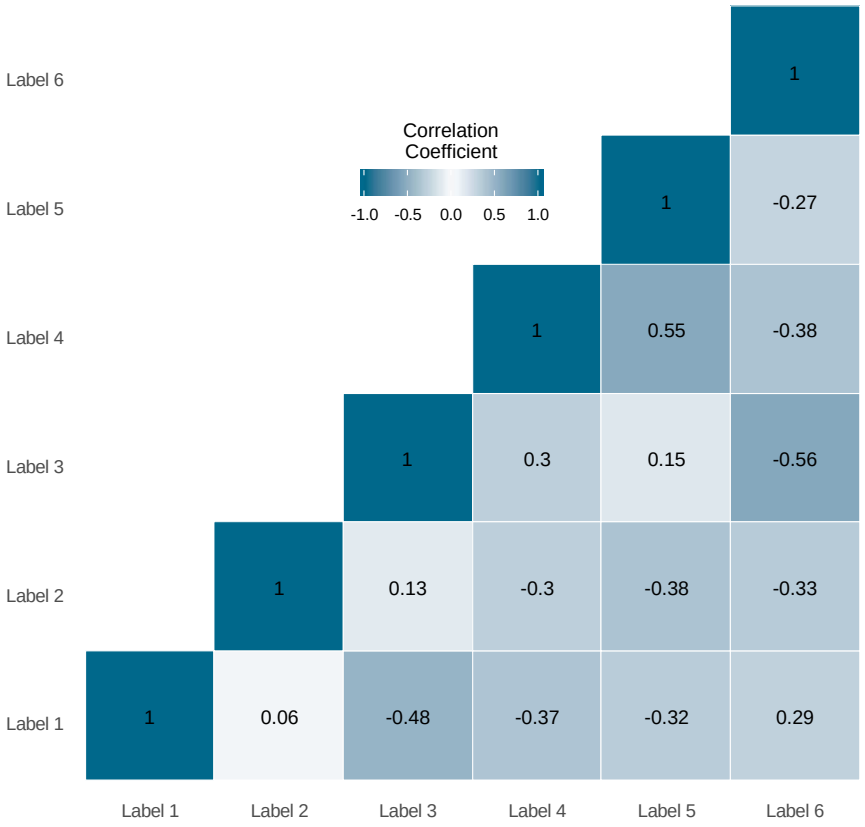


Figure 4.2 Heatmap representing the label correlations for the *Emotions* dataset.

The features of the *Emotions* dataset fall into two categories: *rhythmic* and *timbre*. For more detail on the creation of these features, the interested reader is referred to Section 4.1 of Trohidis *et al*. (2008). The 72 features are listed in Appendix A[1].

In the next section the experimental approach that is followed to analyse the efficiency of the proposed FS procedure on the *Emotions* dataset, is discussed.

[1] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

4.4 Experimental approach

There has recently been a significant increase in the literature available on FS in the multi-label context. In many cases, these techniques have mainly been tested and compared using the available benchmark datasets. The main objective of the empirical study in this chapter is to present the results obtained if the proposed FS procedure, RPFS, is applied to the *Emotions* dataset. The two classification algorithms used are SVMs and XGBoost. These classifiers are compared using the *Emotions* dataset and the results are presented in Section 4.5.

An extensive empirical study was performed on the *Emotions* dataset to achieve the objective mentioned above. The procedure can be outlined as follows:

1) Randomly split the data into a training dataset (70%) and a test dataset (30%).
2) Apply steps (i) – (iii) below on the training dataset generated in 1).
   i) Construct a relevance matrix representing the associations between features and labels.
   ii) Perform MCA on the relevance matrix.
   iii) Construct an MCA biplot and RPFS plot.
3) Perform FS, including the following sets of features:
   d) Select only the features identified as relevant features.
   e) Select the top ranked feature from each of the relevant feature groups.
   f) Select the two top ranked features from each of the relevant feature groups.
4) Use binary relevance and:
   a) an SVM, and
   b) XGBoost classifier.
5) Calculate the four multi-label evaluation measures (Hamming-loss, Precision, Recall, and One-error) on the test data.
6) Repeat 1) to 5) 100 times and calculate the averages of the evaluation measures over these repetitions.

Various graphical representations such as bar charts and box plots will be used to compare the performance of the different relevance measures, FS techniques and two classifiers employed.

In this section, the experimental approach that was followed, is discussed. The more detailed discussion of this procedures is presented in the next sections.

4.4.1 Constructing the relevance matrix

As mentioned in Section 3.3.2, the three relevance measures, namely Information Gain (IG), ReliefF, and the correlation coefficient, are each used to construct a relevance matrix that represents the associations between the features and the labels. The threshold values for each of these measures are determined empirically using the SVM classifier (more information on the selection of the thresholds will be provided in the next paragraph). Threshold values from a predetermined interval of values are selected and the ten evaluation measures discussed in Chapter 2 are calculated.

For example, for the IG, cut-off values between 0.01 and 0.3 are used on the original test and train split (as posted on MULAN) of the *Emotions* dataset described in Section 4.3. The ten evaluation measures are then compared to determine the appropriate threshold value. This is done while also taking the number of feature groups created into account. It can be argued that selecting the thresholds in this manner could lead to a subjective selection of the thresholds. Consider Table 4.3 below.

Table 4.3: Threshold selection for *Emotions* dataset using IG to determine relevance.

Evaluation Measure	Full Dataset	Threshold		
		0.01	0.05	0.1
Hamming-loss	0.25660	0.24010	0.24175	0.24340
Classification Accuracy	0.14851	0.15842	0.15842	0.14356
Precision	0.57261	0.58911	0.58746	0.58581
Recall	0.86469	0.88861	0.88366	0.89274
F1-score	0.67013	0.68911	0.68663	0.68779
Accuracy	0.55347	0.57170	0.57087	0.56667
One-error	0.25248	0.25248	0.24257	0.25743
Coverage	0.54043	0.50990	0.51650	0.48350
Ranking loss	0.15451	0.14442	0.14651	0.14370
Average Precision	0.81653	0.82294	0.82515	0.82404
Number of groups		**10**	**17**	**11**

If one considers the Hamming-loss, there is a slight improvement when the threshold IG value decreases from 0.1 to 0.05 to 0.01, but one also notices that these changes in the threshold influence the amount of feature reduction that takes place. In this case, the number of groups provides an indication of how much feature reduction takes place. For the threshold of 0.1, the 72 individual features are reduced to ten (the number of feature groups (eleven) minus the feature group containing the features that are deemed to be irrelevant). Similarly, for a threshold of 0.05, the feature groups are reduced to 16, and for 0.01 to nine. In this case, it would be sensible to select a threshold, such as 0.05, which is more conservative with respect to the number of feature groups, *i.e.* more features are included in the model. This method of selection was used for the determination of all thresholds in this study. These same thresholds are then applied when fitting the XGBoost classifier for the results to be comparable.

For the relevance matrix, features corresponding to entries above the threshold are deemed to be relevant for the corresponding labels, implying that the particular entry in the relevance matrix receives a value of 1. For example, for the *Emotions* dataset a feature is deemed relevant for a given label if the IG value exceeds the threshold 0.05.

The threshold values for the correlation coefficient and for ReliefF are determined in the same manner. For the correlation coefficient an interval between 0.1 and 0.5 was used to determine the optimal threshold. A feature is deemed to be relevant for a specific label if the absolute correlation between the feature and the label exceeds 0.2.

The two parameters for ReliefF are the number of ReliefF iterations, m, and the significance level, α. A threshold value, τ, is calculated as $1/\sqrt{\alpha m}$. The w-values obtained using ReliefF (refer to Section 3.3.2) are then compared to the value of τ and a feature with a w-value larger than τ is considered relevant. The choice of number of ReliefF iterations is set to 10 000. This decision is based on the fact that 10 000 is the default number of iterations for the procedure employed in R. Other values of m considered were 5 000 and 2 000, but even with 5 000 iterations, very few features were deemed to be relevant. For the significance level, α, the values 0.005, 0.01, 0.025, 0.05, and 0.1 were considered. For values smaller than 0.05 only a few features were deemed to be relevant. There is very little difference between the number of features deemed to be relevant if the significance level is set to 0.05 or 0.1. The default value of 0.05 was therefore applied for the *Emotions* dataset.

The application of RPFS on the relevance matrix described in this section is discussed in the next section.

4.4.2 Performing feature selection using relevance pattern feature selection

The MCA biplot which provides a graphical representation of the different feature groups, are obtained by using the R libraries `UBbipl` and `UBFigs` (Le Roux and Lubbe, 2013). The MCA biplots and RPFS plots are independent of the classifier and the plots in Figures 4.3 to 4.8 are used to construct the feature groups used for both the SVM and XGBoost classifiers.

The features in each feature group are ranked according to either the absolute correlation, the IG values or the w-values obtained from ReliefF.

As mentioned in Section 3.4.5, four different models are proposed based on RPFS. The full model is included in order to provide a benchmark performance to compare the FS procedures against.

1. *Full*: All p features are included, *i.e.* no FS is performed.
2. *Relevant*: Remove the group of features identified to be irrelevant. Let $\hat{k}$ = the estimated number of irrelevant features. Select only the $p - \hat{k}$ "relevant" features. The term "relevant" here refers to the features that are identified to be relevant by the relevance measure, *i.e.* IG, absolute correlation coefficient, or ReliefF.
3. *Highest*: Select only the top ranked feature from each of the relevant feature groups. Features are ranked according to the values obtained from the relevance measures, *i.e.* the IG values, the absolute correlation coefficients, or the w-values for ReliefF.
4. *Highest 2*: Select only the top two ranked features from each of the relevant feature groups.

For illustration purposes, Figures 4.3 to 4.8 are constructed based on the original training and test split of the *Emotions* dataset which is available on the MULAN website.

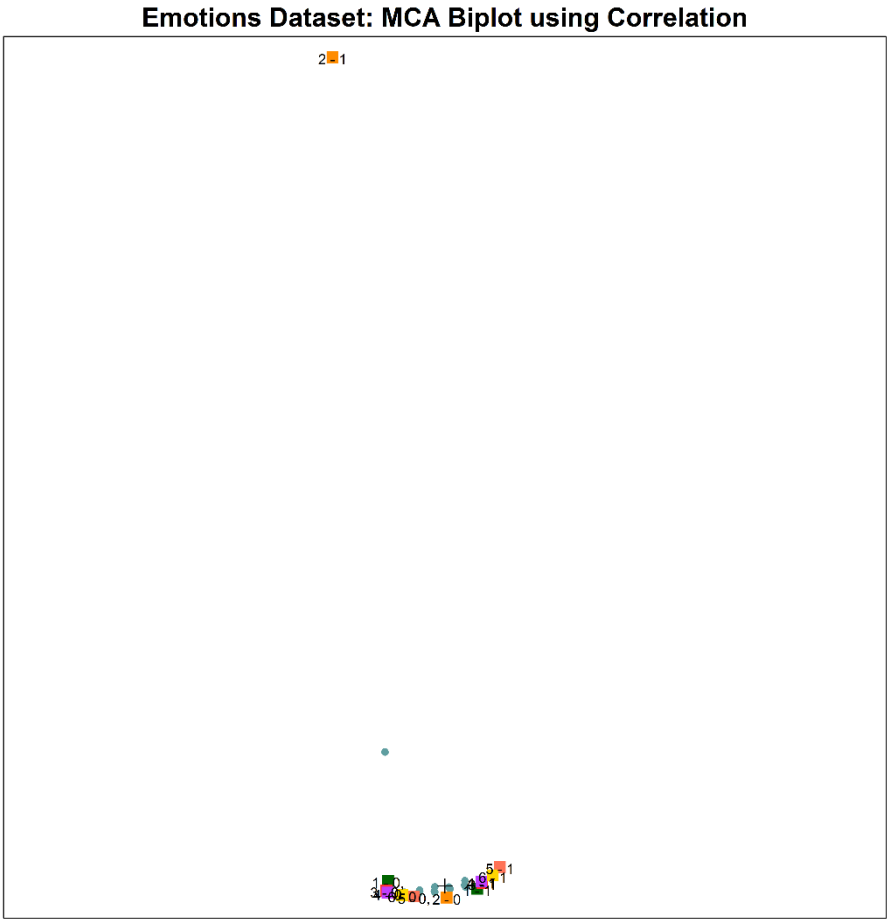


Figure 4.3 MCA biplot using the correlation coefficient as relevance measure.

The 72 features are all plotted in the MCA biplot above. Features (and labels) that are plotted close together in the biplot are similar in terms of the information that they provide and features (and labels) that plot further from each other are dissimilar. Figure 4.3 shows the MCA biplot for the case where the correlation coefficients between the 72 features and the six labels are used to determine feature relevance. From Figure 4.3 it is interesting to note that label 2 (*Amazed-Surprised)* plots much further from the other labels and that a single feature group also plots further from the other feature groups. This feature group represents all features that are deemed relevant only for label 2.

The column points in this figure make it difficult to distinguish between the distinct feature groups. For this reason, the RPFS plot in Figure 4.4 is preferred when the aim is to perform FS.

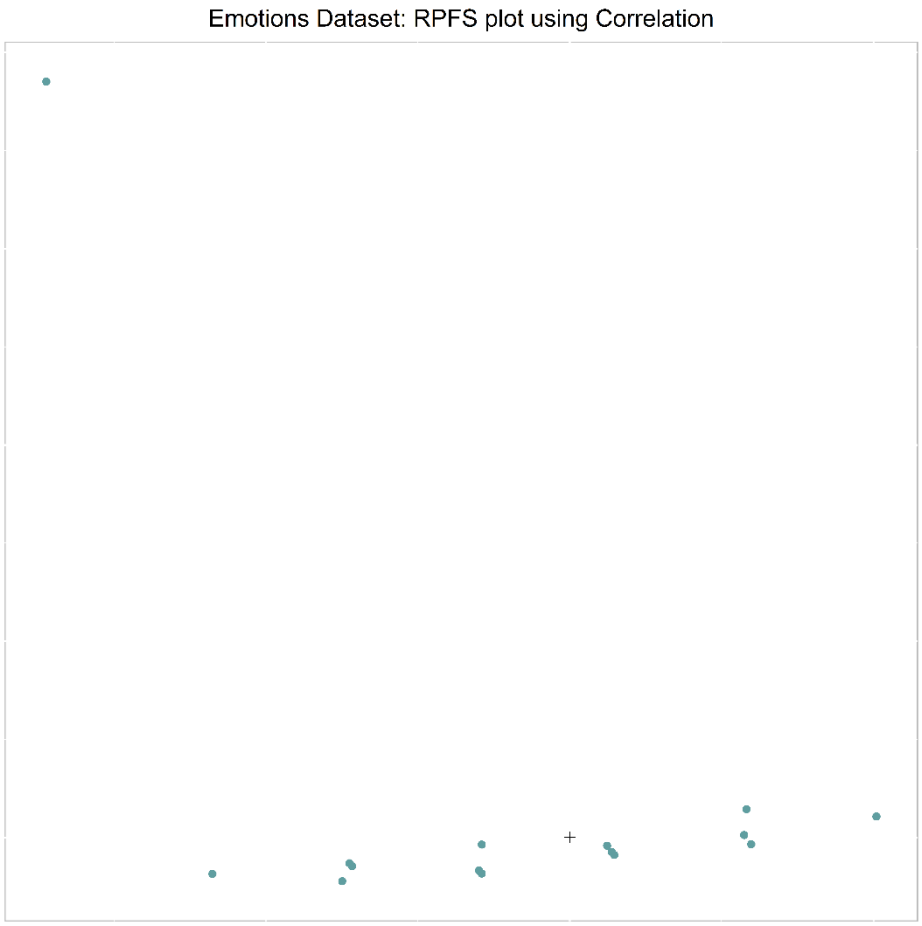


Figure 4.4 RPFS plot using the correlation coefficient as relevance measure.

A number of clear, distinct feature groups emerge from the RPFS plot. The 72 features are all plotted in the RPFS plot above. Features that are plotted close together in the plot are similar in terms of the information that they provide and features that plot further from each other are dissimilar. Figure 4.4 shows the RPFS plot for the case where the correlation coefficients between the 72 features and the six labels are used to determine feature relevance. From Figure 4.4, one is able to distinguish between 15 distinct feature groups. A more detailed discussion on, for example, the number of features in each of these groups will be given later in this section.

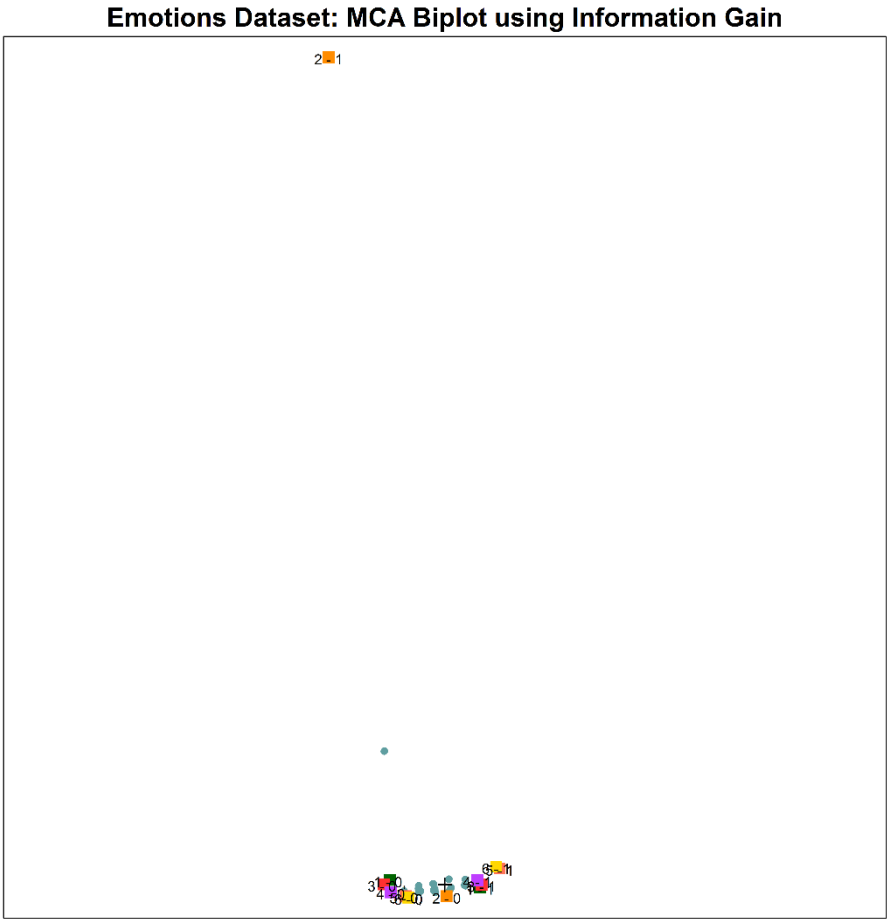


Figure 4.5 MCA biplot using IG as relevance measure.

Figure 4.5, based on calculating the IG values between the 72 features and the six labels, provides a similar MCA biplot to that seen in Figure 4.3. Label 2 (*Amazed-Surprised)* again plots much further from the other labels and the feature group representing all features that are deemed relevant only for label 2 also plots further from the other feature groups.

The RPFS plot in Figure 4.6 only includes the row points from the MCA biplot, allowing for a visual representation of the feature groups. Seventeen unique feature groups can be identified in Figure 4.6.

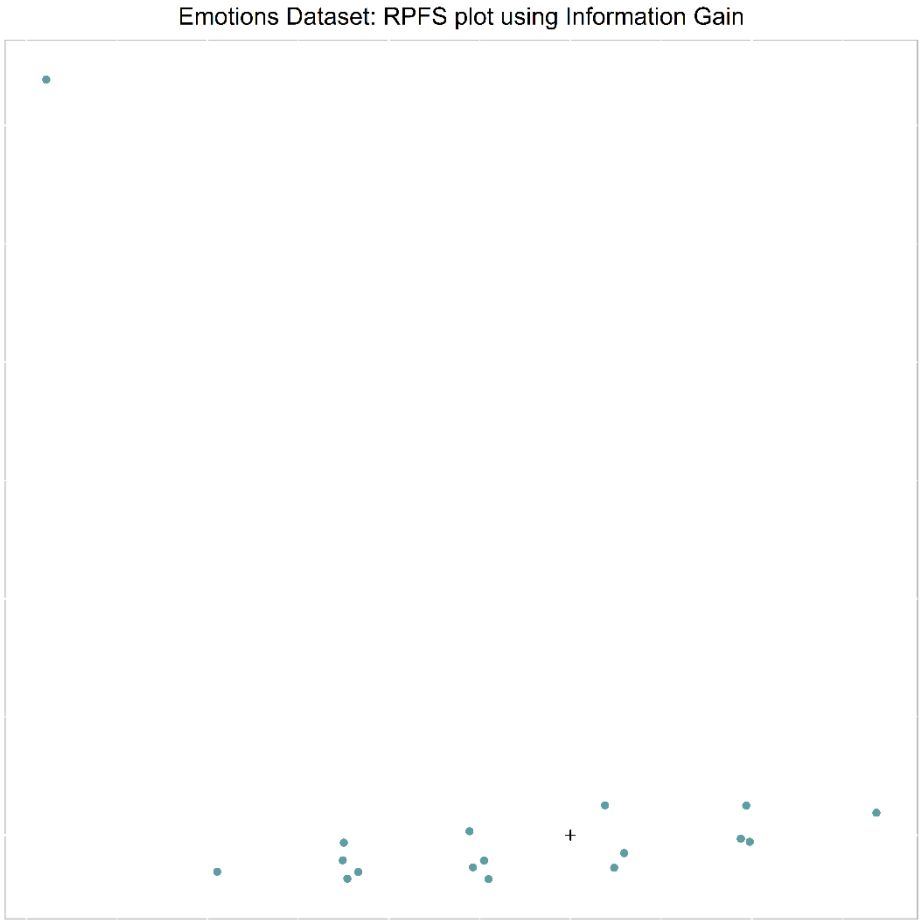


Figure 4.6 RPFS plot using IG as relevance measure.

In Figures 4.7 and 4.8 the feature groups based on a relevance matrix obtained using ReliefF as relevance measure are shown in the MCA biplot and RPFS plot, respectively. Figure 4.7 provides interesting insights into the relationship amongst the labels and the features.

The first interesting observation is that all the column points that indicate the presence of a label, plot on the right side of the biplot. If one isolates the ten feature groups on the right side of the MCA biplot, one finds that these ten feature groups all include features that are relevant for either four, five or six labels.

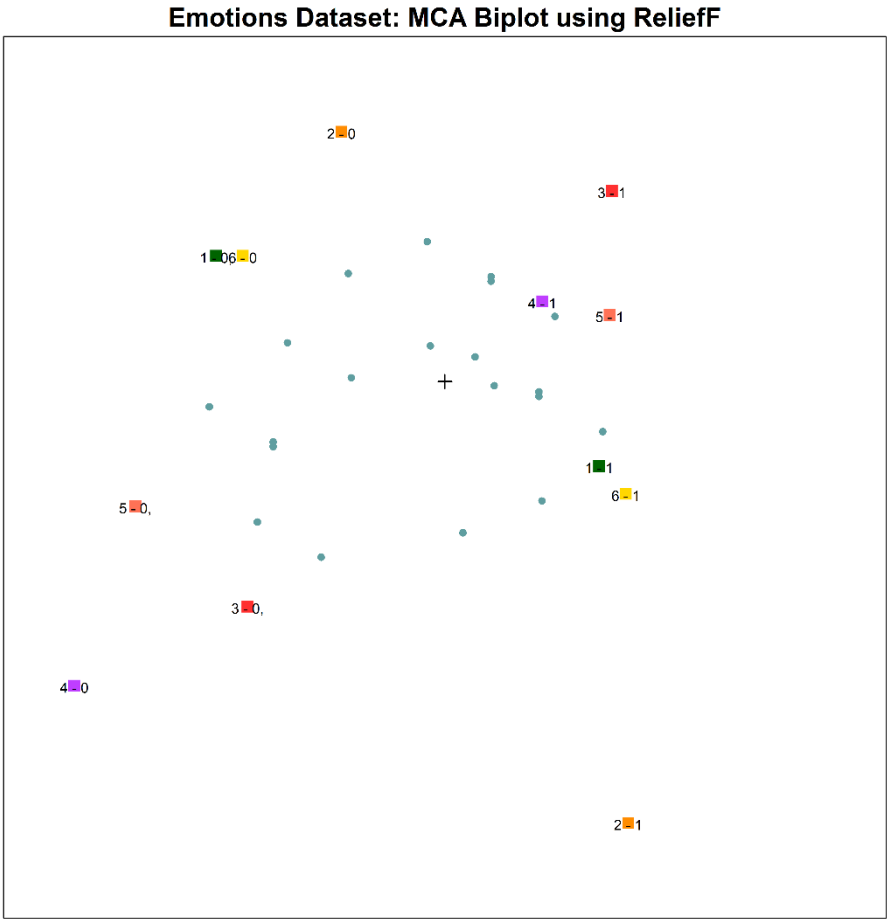


Figure 4.7 MCA biplot using ReliefF as relevance measure.

The second interesting observation is that the feature group that contains the features that are deemed to be irrelevant is the group that plots furthest to the left on the MCA biplot.

In the RPFS plot in Figure 4.8, twenty unique feature groups are formed using ReliefF as relevance measure.

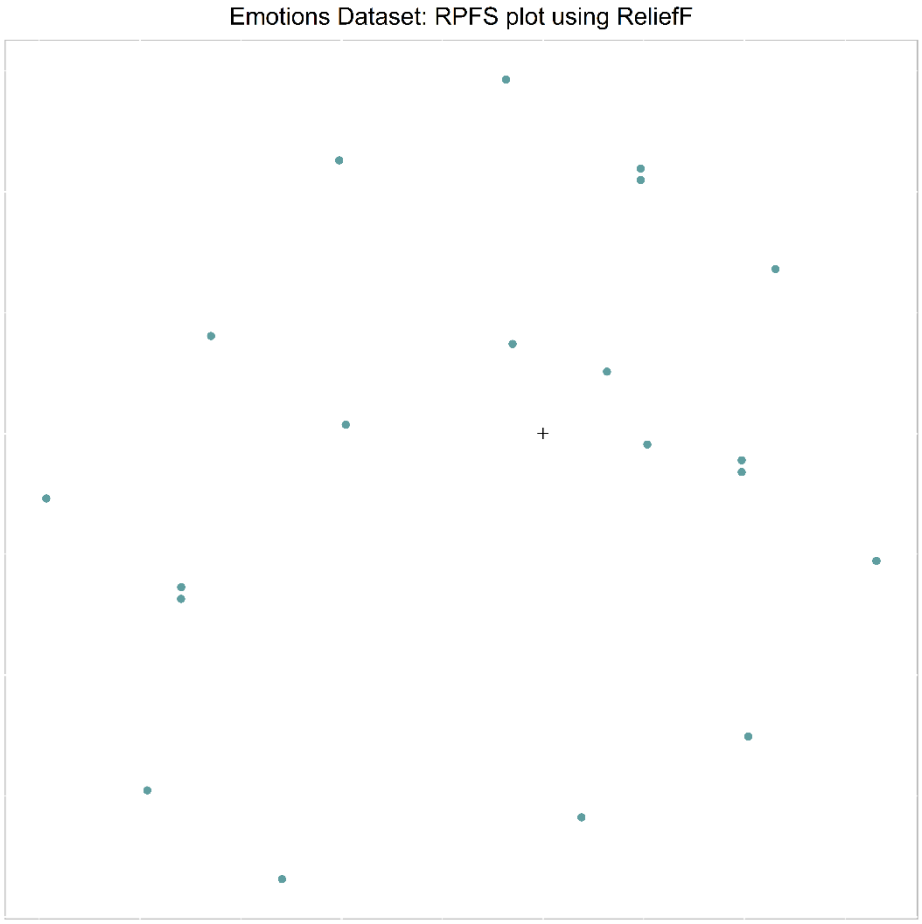


Figure 4.8 RPFS plot using ReliefF as relevance measure.

While the RPFS plots based on the correlation coefficient and IG look similar, the plot for ReliefF seems to very different. It is important to note that the feature group in the top left-hand corner of Figures 4.4 and 4.6 causes the scales of the plots to differ, and caution should be taken in interpreting the differences between the plots in Figures 4.4, 4.6, and 4.8 without considering the scale. If the feature group in the top left-hand corner (this feature group represents all features that are relevant only for label 2) is excluded from Figures 4.4 and 4.6, Figure 4.9 is obtained. When the one outlying feature group is removed, the three plots do not look too different

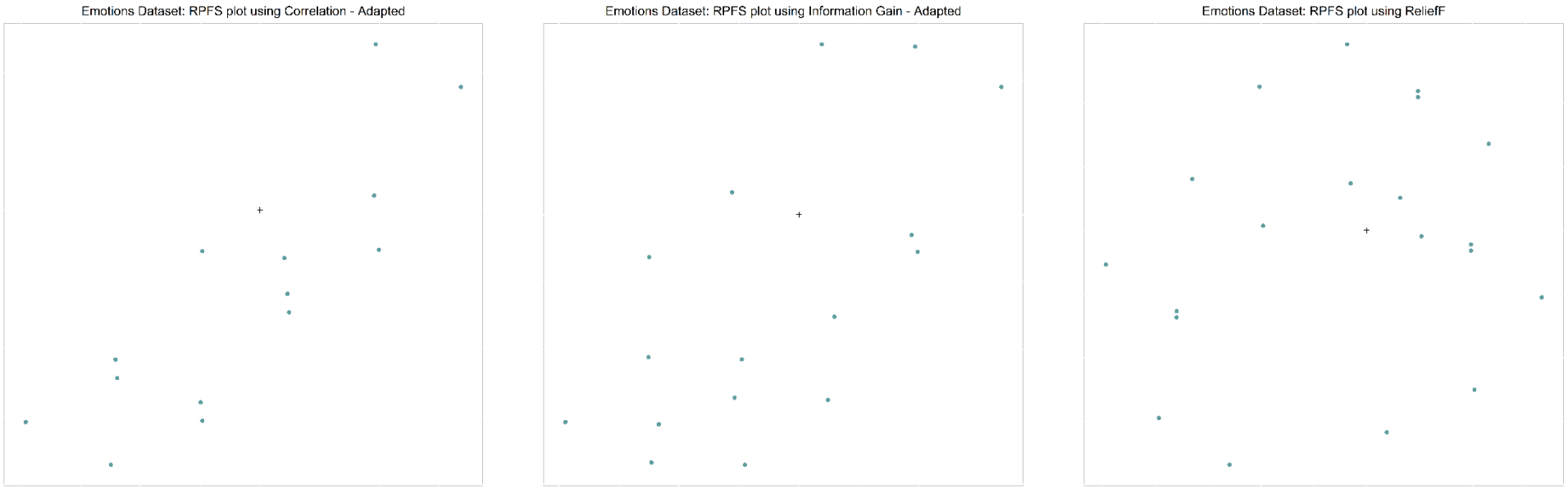


Figure 4.9 Comparison of feature groups.

The outlying feature group contains only a single feature, namely feature 8. Feature 8 is the mean of the fourth MFCC calculated over all frames (*Mean_Acc1298_Mean_Mem40_MFCC_4*). It will be shown later that this feature is one of two features deemed relevant by all three relevance measures. It is interesting to note that both the correlation and IG consider this feature to be dissimilar from the other features, while ReliefF does not.

If features that are plotted *close* together are similar, it would be reasonable to assume that these features will provide similar information to the classifier. How close is *close*? In this dissertation, a strict definition of *close* is applied by grouping features together only if these groups lie directly on top of each other, *i.e.* the coordinates are identical. While this constraint is strict, it is important to note that this ensures a more conservative number of feature groups (*i.e.*, a larger number of features will be included in the models that are subsequently fitted).

The sizes of the feature groups are presented in Figures 4.10 to 4.12. The category labels present the label relevance of the group, for example in Figure 4.10, the largest feature group is the group in which the features are relevant for all labels except for label 2, *i.e.* $\begin{bmatrix}1 & 0 & 1 & 1 & 1 & 1\end{bmatrix}$; the second largest feature group is the group that contains all the irrelevant features, *i.e.* $\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$, and so on. Feature groups that are empty, *i.e.* that do not contain any features are omitted from the bar chart. For example, no features are assigned to the feature group $\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$.

In Figure 4.7 the 72 features are grouped into a total of only 15 feature groups. Notably, using the correlation coefficient as relevance measure leads to no features that are considered to be globally relevant, *i.e.* relevant for all the labels. There are 56 locally relevant features and 16 features are deemed irrelevant.

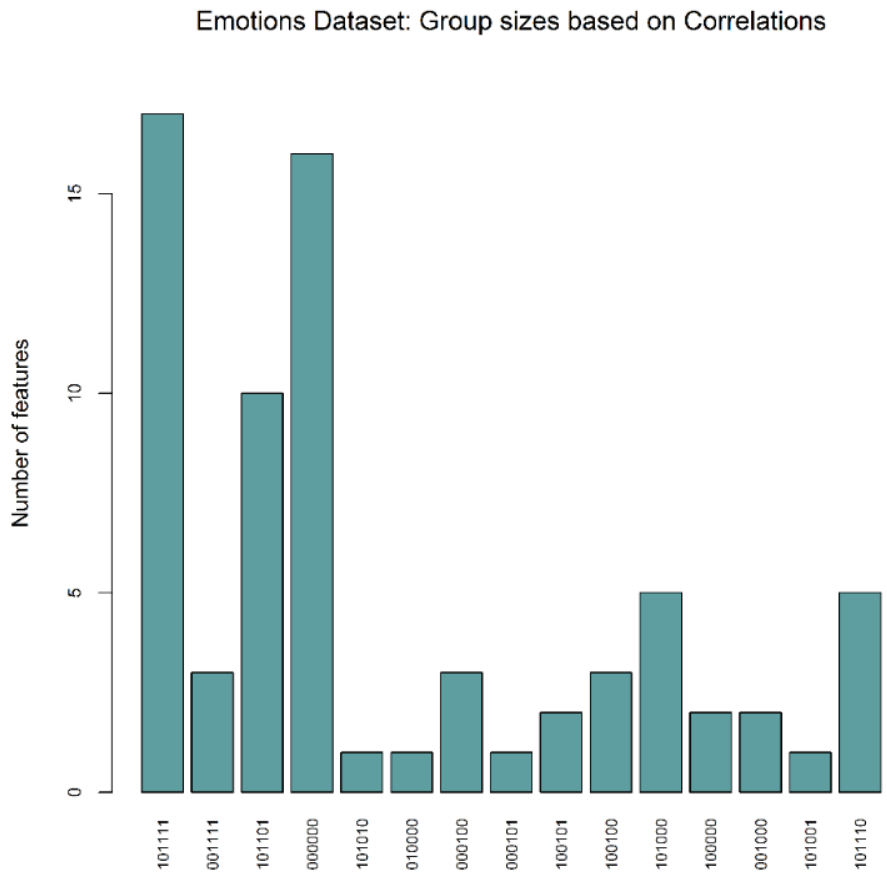


Figure 4.10 Feature group sizes using the correlation coefficient as relevance measure.

The number of features used for each of the four previously defined FS procedures is as follows:

1. *Full:* All 72 features are included.
2. *Relevant:* Select only the 56 features identified as relevant based on the correlation coefficient.
3. *Highest:* Select the top ranked feature from each of the 14 relevant feature groups.
4. *Highest 2:* Select the top two ranked features from each of the 14 relevant feature groups. In this case, that leads to 24 features being selected.

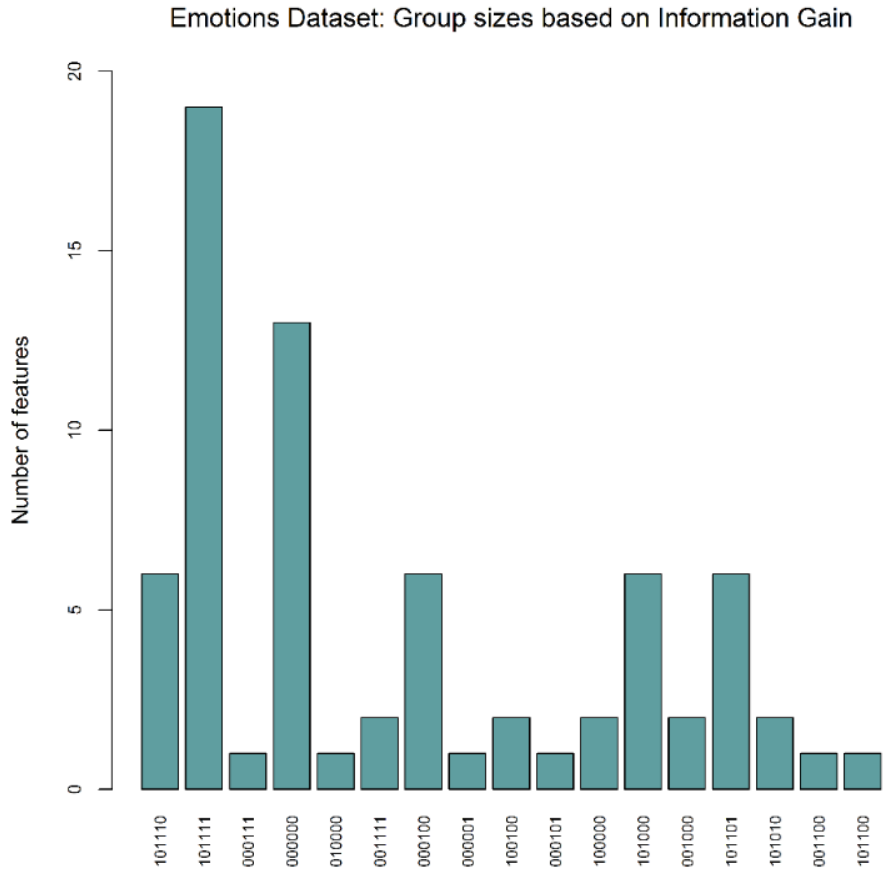


Figure 4.11 Feature group sizes using IG as relevance measure.

Based on IG as relevance measure, the 72 features are grouped into a total of only 17 feature groups. The resulting grouping differs from that obtained using the correlation coefficient, but once again no features are considered to be globally relevant. There are 59 locally relevant features, while 13 features are deemed irrelevant.

The number of features used for each of the four FS procedures is as follows:

1. *Full:* All 72 features are included.
2. *Relevant:* Select only the 59 features identified as relevant based on IG.
3. *Highest:* Select the top ranked feature from each of the 16 relevant feature groups.
4. *Highest 2:* Select the top two ranked features from each of the 16 relevant feature groups. In this case, that leads to 26 features being selected.

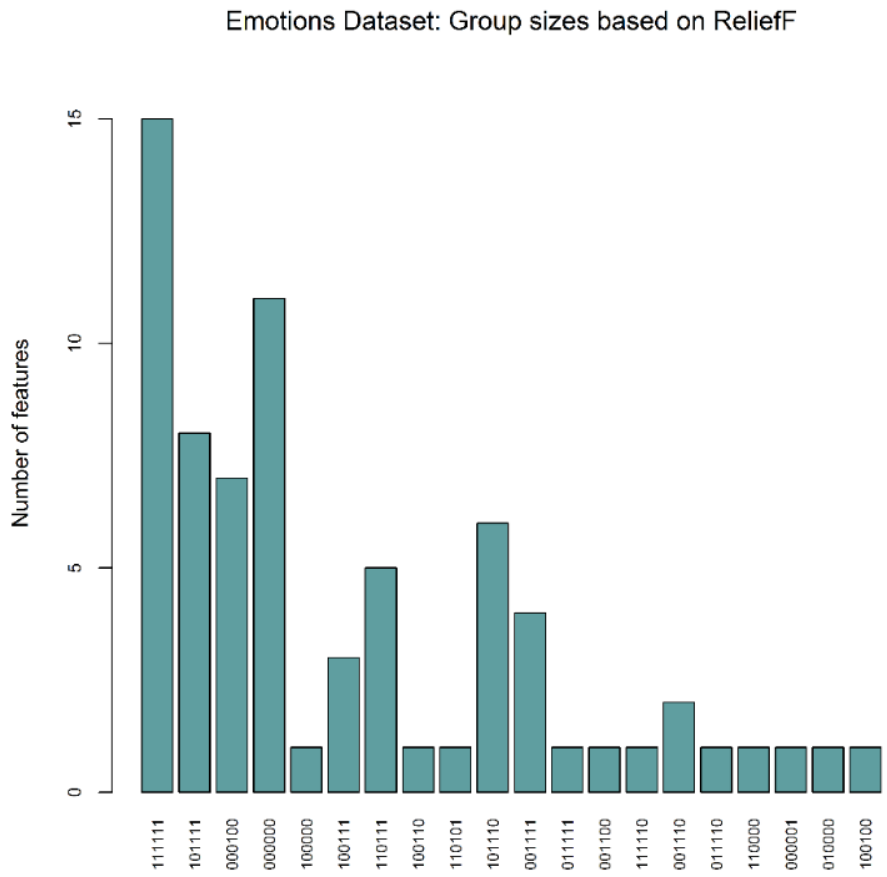


Figure 4.12 Feature group sizes using ReliefF as relevance measure.

Based on ReliefF as relevance measure, the 72 features are grouped into a total of only 20 feature groups. The resulting grouping shows a number of larger groupings which contain five or more features and then eleven feature groups that contain only a single feature. ReliefF deemed 15 features to be globally relevant. There are 61 locally relevant features, while 11 features are deemed irrelevant.

The number of features used for each of the four FS procedures is as follows:

1. *Full:* All 72 features are included.
2. *Relevant:* Select only the 61 features identified as relevant based on ReliefF.
3. *Highest:* Select the top ranked feature from each of the 19 relevant feature groups.
4. *Highest 2:* Select the top two ranked features from each of the 19 relevant feature groups. In this case, that leads to 27 features being selected.

In order to provide some insight into the similarities and differences between the results obtained from the different relevance measures, Tables 4.4 and 4.5 are constructed. In Table 4.4, the features deemed to be irrelevant by each of the relevance measures are presented. The features that are shaded green are the features judged to be irrelevant by all three measures. There are seven of these features on which all three relevance measures agree.

Table 4.4: Comparison of features deemed to be irrelevant.

Feature	Correlation Coefficient	IG	ReliefF
6	×	×	
7		×	×
9	×	×	×
10	×	×	
11	×	×	×
12	×	×	
13	×	×	×
14	×	×	×
15	×	×	
16	×	×	×
21	×		
33	×	×	×
34	×		
35	×	×	×
49			×
51	×		
65			×
68	×		
69	×	×	
71			×

These seven features are all part of the category *timbre* features:

- *Mean_Acc1298_Mean_Mem40_MFCC_5*;
- *Mean_Acc1298_Mean_Mem40_MFCC_7*;
- *Mean_Acc1298_Mean_Mem40_MFCC_9*;
- *Mean_Acc1298_Mean_Mem40_MFCC_10*;
- *Mean_Acc1298_Mean_Mem40_MFCC_12*;

- *Std_Acc1298_Mean_Mem40_Centroid*; and
- *Std_Acc1298_Mean_Mem40_Flux*.

The first five of these features are the means of the first 13 Mel frequency cepstral coefficients (MFCCs) calculated over all frames. MFCCs are derived by dividing the signals into frames and calculating the amplitude spectrum for each frame. The logarithm is then taken and converted to Mel scale. For more information on MFCCs refer to Sandrock (2013: 34). Some of the timbre features are extracted from the Short-Term Fourier Transform (FTT): spectral centroid, spectral rolloff, and spectral flux. The last two features, 33 and 35 are the standard deviations of the means of the centroid and flux calculated over all frames. For more information on these features, refer to Trohidis *et al*. (2008).

It is interesting to note that none of the *rhythmic* features (features 65 – 72) were deemed irrelevant by all three measures. The rhythmic features are derived by extracting periodic changes from a beat histogram (Trohidis *et al*., 2008). The two highest peaks are identified using an algorithm that utilises autocorrelation. The amplitudes of the two peaks, their beats per minute (BMPs), and the high-low ratio of the BPMs are calculated. The remaining three rhythmic features are calculated by summing the three histogram bins based on the BPMs.

Using either the correlation coefficient or IG as relevance measure leads to fewer features being selected for the FS procedures. The features ranked highest in each group by each of the three relevance measures are given in Table 4.5. Features 8 and 23 are deemed to be important by all three measures. These are shaded green. Feature 8 is the mean of the fourth MFCC calculated over all frames (*Mean_Acc1298_Mean_Mem40_MFCC_4*). Feature 23 is the standard deviation of the second MFCC over all frames (*Mean_Acc1298_Std_Mem40_MFCC_2*).

Table 4.5: Comparison of highest ranked features.

	Highest Ranked Feature																		
Correlation Coefficient	4	5	7	8	18	19	22	23	25	29	31	50	57	65					
IG	1	3	8	17	21	23	24	26	32	43	47	56	57	66	67	70			
ReliefF	2	8	18	20	22	23	26	34	43	46	50	52	56	64	66	67	69	70	72

It is of interest to compare the different FS approaches in terms of their sparsity, *i.e.* in terms of the number of features retained by these procedures.

Spolaôr *et al*. (2013) note that the *Feature Reduction* measure defined below can be used to evaluate the average reduction in the number of features identified by each FS procedure:

$$\text{Feature Reduction } (D, X') = 100 - \frac{100 \mid X' \mid}{p}$$

where X' is a subset of the features from dataset D with p features.

The average Feature Reduction percentages over SVM and XGBoost for the FS procedures using only the relevant and the highest ranked feature per feature group are provided in Table 4.6. These values are determined using 100 splits of the *Emotions* dataset.

Table 4.6: Feature Reduction percentages.

Technique	Percentage
Correlation Relevant	26.24
Correlation Highest	80.42
IG Relevant	27.51
IG Highest	79.26
ReliefF Relevant	12.04
ReliefF Highest	71.08

For example, the technique that utilises IG which includes all features that are judged to be relevant, reduces the number of features by 27.51%. This means that the final model only uses 52 features ($(1 - 0.2751) \times 72$). The number of features used for the procedures based on the correlation coefficient and IG is very similar, but some differences exist for those based on ReliefF.

The problem transformation method and the classification techniques that were applied to the *Emotions* dataset following FS are discussed in the next section.

4.4.3 Classification

In this section, the classification approach used in the empirical investigation in Chapter 4 is discussed. The multi-label *Emotions* dataset is split between a training (70%) and test (30%) set and is then transformed to single-label datasets using the binary relevance (BR) approach. The single-label classifiers, SVM and XGBoost, are then applied to these transformed datasets, and the predicted labels are compared to the true labels of the test dataset. The ten multi-label evaluation measures defined in Section 2.3 are then calculated. The split between training and test steps are repeated 100 times and the mean and the median for each of the evaluation measures are calculated over the 100 repetitions. Due to the presence of outliers for some techniques, the results are also presented in boxplots. These representations allow for the comparison of methods based on location and variation. In the remainder of this section, a brief discussion of the specific algorithms used will be presented.

Binary Relevance

The choice of BR as problem transformation measure is based on two considerations: firstly, the popularity of BR in related research, and secondly the fact that FS in the multi-label context is a complex enough problem as it is. If a new multi-label FS approach can be found that performs well when using BR, the technique can be extended to other problem transformation techniques or algorithm adaptation approaches.

Support vector machines

In the results reported below, the SVM was fitted using a radial basis function (RBF) kernel $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = e^{-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}$ (as defined in Section 2.7.1) with hyperparameter $\sigma = 1/p$, where p refers to the number of features. The cost parameter of the SVM, denoted by C, was chosen as $C = 1$. The choice of both the hyperparameter and the cost parameter is motivated by the empirical results obtained by Sandrock (2013). The author finds that examination of the classification results indicates that values of $C = 0.1$ and $C = 1$ seem to give the best results for the *Emotions* dataset. The final choice of $C = 1$ for the studies reported below is based on the fact that this value is proposed at the default value in the literature.

Extreme gradient boosting

The important parameters associated with XGBoost are the objective function, the number of iterations/passes on the data, the step size of each boosting step, and the maximum depth of the trees. Using logistic regression for classification as the objective function returns a predicted probability, not a class. In this study, a predicted probability larger than 0.5 is used to predict the presence of the response variable (label).

The other three parameters are determined using the cross-validation function of XGBoost. These parameters are estimated by minimising the mean test error during cross-validation. The step size of each boosting step, η, typically ranges between 0 and 1. In this study, the cross-validation was performed using the values 0.3, 0.5, and 0.7 for η. Typically, the maximum depth of the tree is less than 10. This parameter is used to control over-fitting at a higher depth. During cross-validation, the values 1, 2, and 4 are used. Finally, the number of iterations or the number of passes on the data are based on the minimum values of the mean test error. The number of passes used during cross-validation was set to the default of 100 passes.

It is important to note that the main aim of this dissertation focusses on developing a new multi-label FS technique. As such, optimisation of the classification algorithms used in this study is not of primary importance. The results of the empirical investigation are presented and discussed in Section 4.5.

4.5 Results and conclusions

In this section, the results using the SVM classifier based on the three relevance measures will be presented. A summary of the medians for each of the ten multi-label evaluation measures introduced in Section 2.3 will be provided. The four evaluation measures used in this study (Hamming-loss, One-error, Precision, and Recall) are summarised in boxplots and discussed. Due to the difficulties associated with displaying the results for all ten multi-label evaluation measures, four measures were selected. Three evaluation measures were selected from the six example-based evaluation measures: *Hamming-loss*, *Precision*, and *Recall*. Only one measure was selected from the four ranking-based evaluation measures: *One-error*. In Section 4.5.2 these graphical representations are repeated for the results obtained using the XGBoost classifier. Finally, the two classifiers are compared in Section 4.5.3.

4.5.1 Relevance pattern feature selection - Support vector machine classifier

In this section, the performance of the FS procedures based on RPFS is evaluated when the SVM classifier is used. The ten evaluation measures described in Section 2.3 are calculated for each one of the 100 iterations and the medians of the measures are calculated and shown in Tables 4.7 to 4.9. Cells that are shaded pink represent FS procedures that perform better than the full set of features. For each iteration, the number of feature groups is recorded. This is then averaged over the 100 iterations and is included in each of the tables below.

Table 4.7: Summary of results using the correlation coefficient as relevance measure (SVM).

	Full	Relevant	Highest ranked	First two ranked
Hamming-loss	0.27247	0.27294	0.28371	0.28184
Classification Accuracy	0.11236	0.11798	0.11236	0.11236
Precision	0.54120	0.54026	0.52715	0.53184
Recall	0.86517	0.86517	0.84738	0.85487
F-one	0.64494	0.64242	0.62931	0.63474
Accuracy	0.51999	0.51985	0.50599	0.50890
One-error	0.23596	0.24719	0.28090	0.26404
Coverage	0.55337	0.56648	0.61564	0.59316
Ranking loss	0.15140	0.15311	0.17061	0.16213
Average Precision	0.82034	0.81351	0.79282	0.80276
Average number of groups	**15.1**			

Table 4.8: Summary of results using IG as relevance measure (SVM).

	Full	Relevant	Highest ranked	First two ranked
Hamming-loss	0.27622	0.27341	0.29073	0.28464
Classification Accuracy	0.11236	0.10674	0.10112	0.10112
Precision	0.53558	0.53558	0.51873	0.52434
Recall	0.86376	0.86517	0.83848	0.85066
F-one	0.63970	0.64167	0.61919	0.63043
Accuracy	0.51381	0.51498	0.49593	0.50454
One-error	0.23596	0.24719	0.28090	0.25562
Coverage	0.56273	0.56601	0.64934	0.61096
Ranking loss	0.15276	0.15591	0.18109	0.16920
Average Precision	0.81676	0.81525	0.78837	0.80020
Average number of groups	**15.8**			

Table 4.9: Summary of results using ReliefF as relevance measure (SVM).

	Full	Relevant	Highest ranked	First two ranked
Hamming-loss	0.27481	0.27434	0.29260	0.28886
Classification Accuracy	0.10674	0.10674	0.10112	0.10112
Precision	0.53652	0.53745	0.51592	0.52247
Recall	0.86704	0.86657	0.83521	0.84363
F-one	0.64176	0.64148	0.61863	0.62453
Accuracy	0.51559	0.51634	0.49363	0.50075
One-error	0.24157	0.24157	0.28652	0.26966
Coverage	0.55758	0.56648	0.64934	0.62125
Ranking loss	0.15313	0.15392	0.18319	0.17395
Average Precision	0.81604	0.81549	0.78674	0.79410
Average number of groups	**21.0**			

The proposed RPFS procedures perform well compared to the full set of features. This is important, since these procedures use substantially fewer features. In some cases, the FS procedure using only the relevant features performs better than the full set of features. A faster learning algorithm is obtained and sometimes even improved performance.

It is important to also consider the amount of variation that is associated with these results. In order to include the variation in the discussion, boxplots for the results of all three relevance measures per evaluation measure are constructed. For the remainder of this section, only Hamming-loss, Precision, Recall, and One-error will be considered. These boxplots are provided in Figures 4.13 to 4.16.

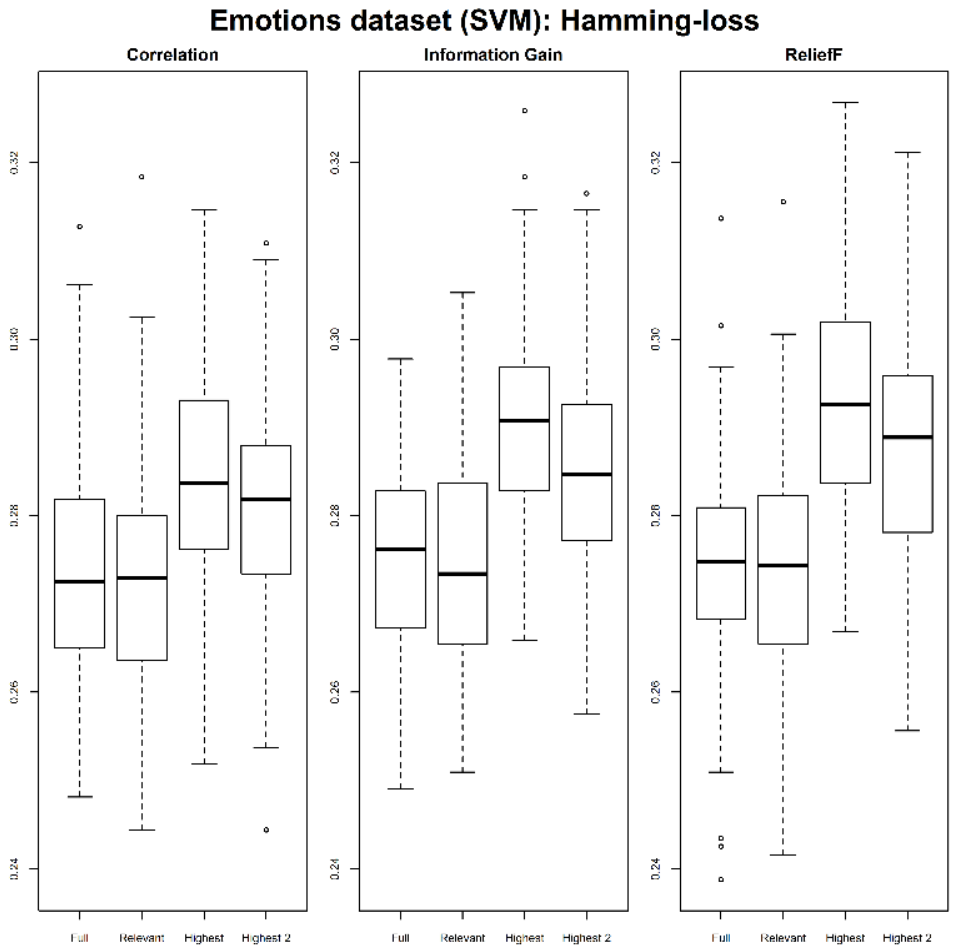


Figure 4.13 Comparison of relevance measures (SVM): Hamming-loss.

For Hamming-loss using IG or ReliefF as relevance measure, the median for the FS procedure using only the relevant features is slightly smaller than the median for the full model. The FS procedures that include fewer features do not perform as well (larger medians), but the results are still fairly similar even though the procedures use substantially fewer features. The boxplots in Figure 4.13 show that the variations in the results of the nine FS procedures are quite similar.

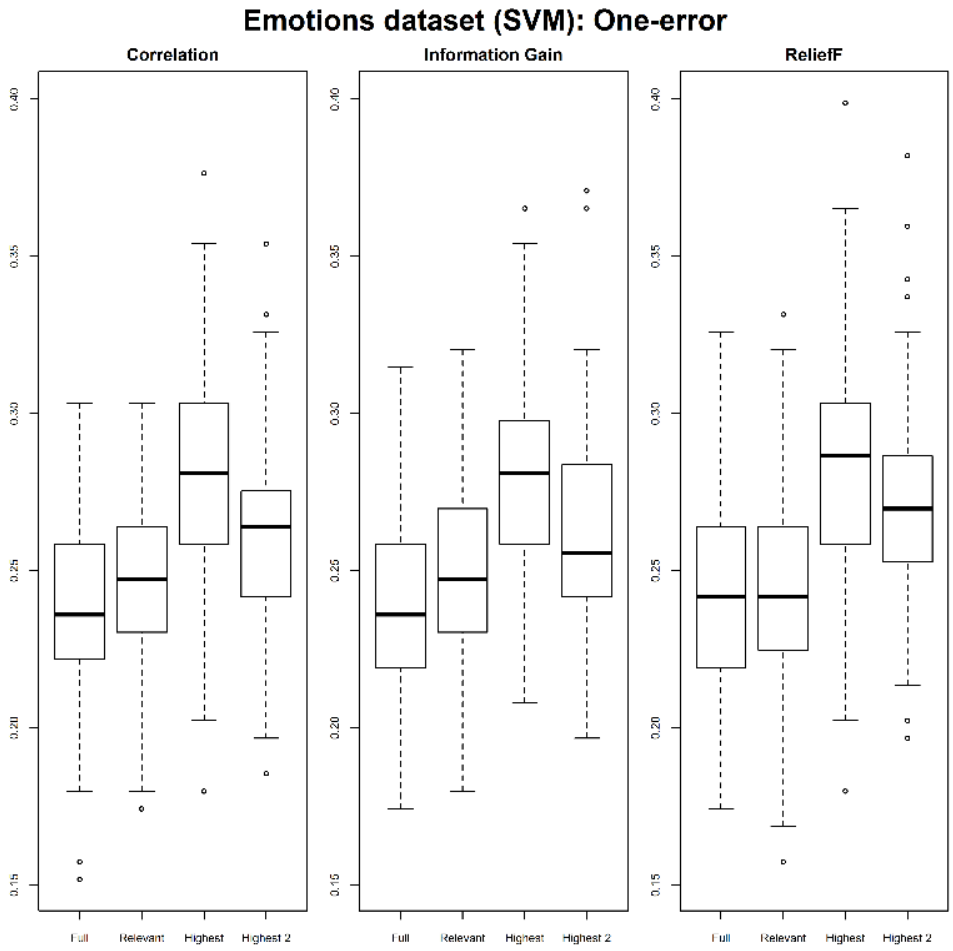


Figure 4.14 Comparison of relevance measures (SVM): One-error.

For ReliefF as relevance measure, the median of the FS procedure using only the relevant features is similar to that of the full model. Once again, the results in Figure 4.14 show similar variation for all nine FS procedures.

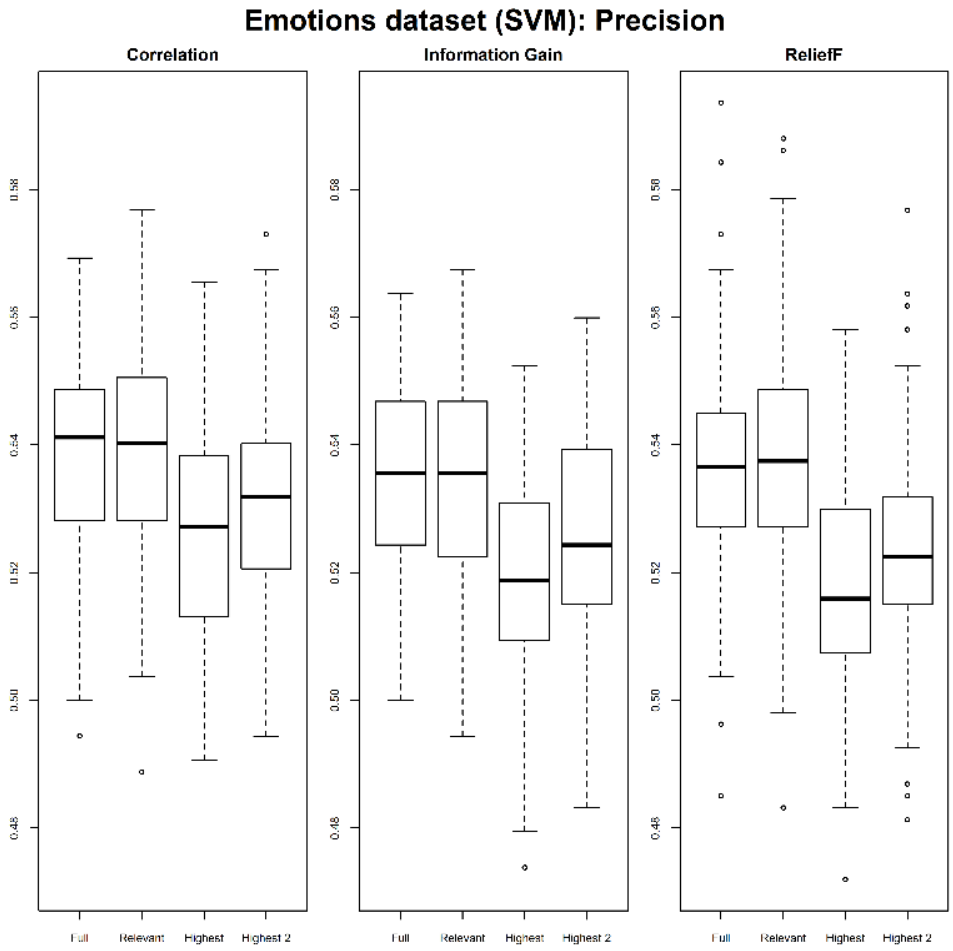


Figure 4.15 Comparison of relevance measures (SVM): Precision.

For Precision in Figure 4.15, the medians of the FS procedure using only the relevant features are similar to that of the full model. The medians of FS procedures that include fewer features are slightly smaller, but the results are still comparable. The correlation relevance measure performs slightly better than IG and ReliefF, with a number of outliers present for the ReliefF relevance measure. The ranges and interquartile ranges of the nine FS procedures are once again similar.

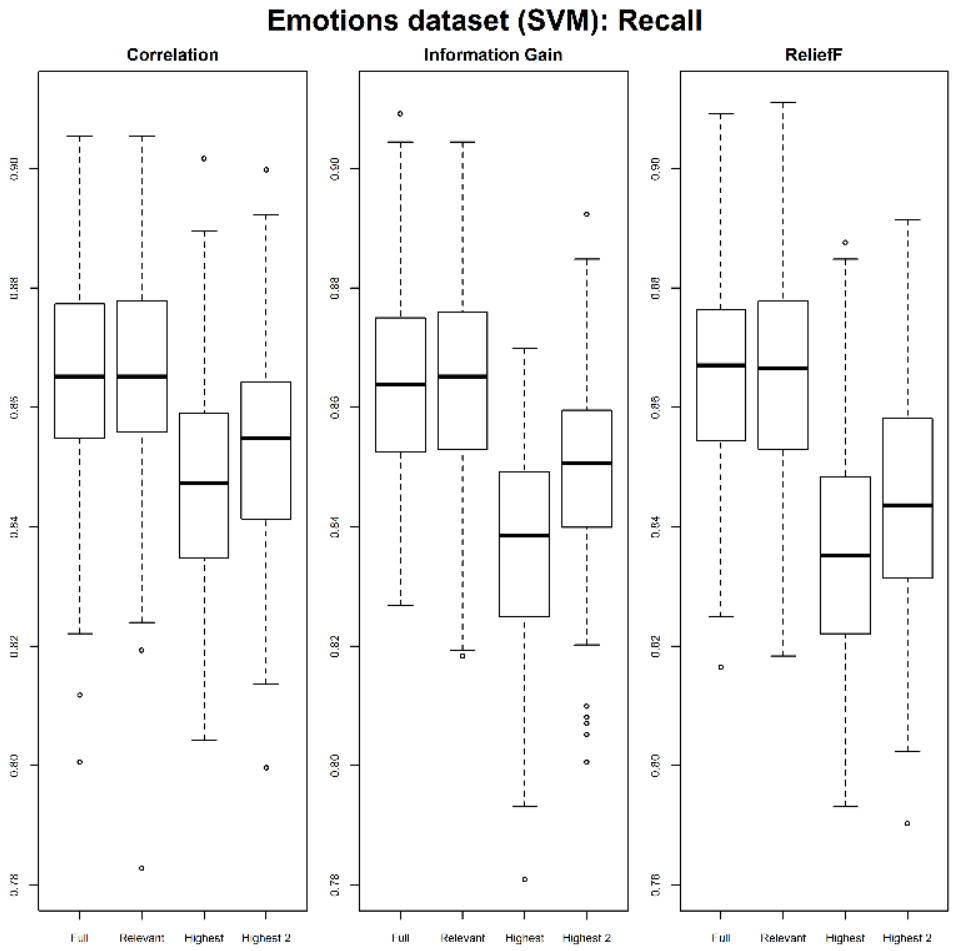


Figure 4.16 Comparison of relevance measures (SVM): Recall.

The medians for Recall shown in Figure 4.16 for the FS procedure using only the relevant features are similar to that of the full model. The medians of the FS procedures that include fewer features are once again smaller, but the results are still of the same order. The correlation relevance measure performs slightly better than IG and ReliefF, with a number of outliers present for the IG relevance measure. The variations of the nine FS procedures are similar.

These boxplots are combined into a single graph (Figure 4.17) to compare the different relevance measures for each of the four evaluation measures. Consider, for example, the results based on the correlation coefficient as relevance measure. For all four evaluation measures, Hamming-loss, Precision, Recall, and One-error, the case where only the features deemed relevant are included performs only slightly better or worse than the full model. The model that includes only the highest ranked features consistently performs worse than the other models. The results based on IG and ReliefF are similar, and the same conclusions can be drawn for each of the models.

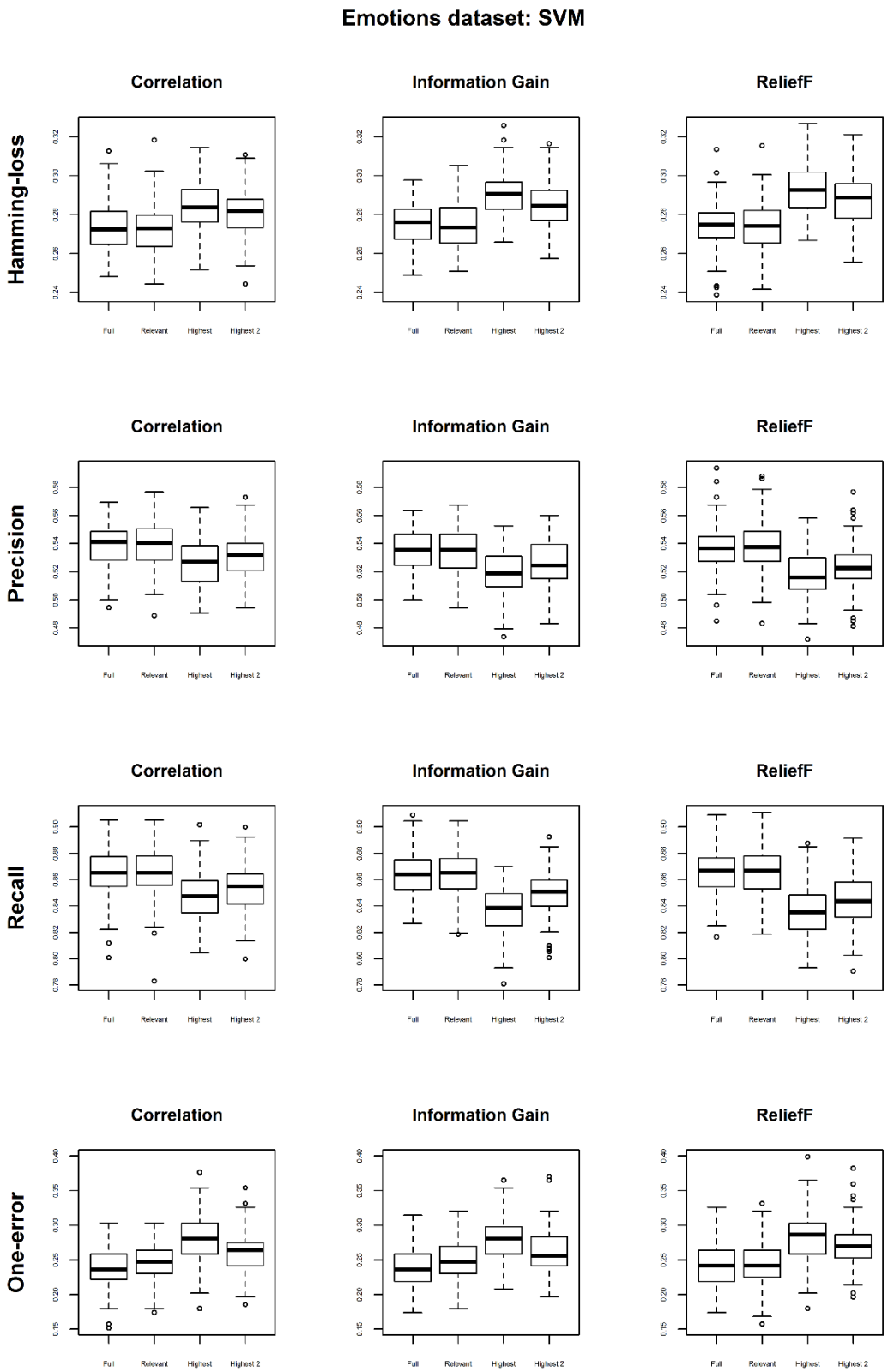


Figure 4.17 Comparison of relevance measures (SVM).

4.5.2 Relevance pattern feature selection - Extreme gradient boosting classifier

The focus of this section is on evaluating the performance of the RPFS procedures when the XGBoost classifier is applied in the same manner as for the SVM classifier in the previous section. All ten evaluation measures are calculated for the 100 iterations and the medians of the measures are calculated and shown in Tables 4.10 to 4.12. Once again, cells that are shaded pink represent FS procedures that perform better than the full set of features.

Table 4.10: Summary of results using the correlation coefficient as relevance measure (XGBoost).

	Full	Relevant	Highest ranked	First two ranked
Hamming-loss	0.19944	0.19897	0.22004	0.21067
Classification Accuracy	0.27247	0.28652	0.25000	0.26404
Precision	0.70858	0.70758	0.66286	0.68388
Recall	0.62734	0.62781	0.58474	0.60534
F-one	0.60609	0.61273	0.56610	0.58839
Accuracy	0.52669	0.53184	0.48689	0.50674
One-error	0.27247	0.27247	0.31461	0.29775
Coverage	0.56976	0.57631	0.67088	0.63530
Ranking loss	0.15826	0.15928	0.19002	0.17626
Average Precision	0.80076	0.80189	0.77178	0.78573
Average number of groups	**15.0**			

Table 4.11: Summary of results using IG as relevance measure (XGBoost).

	Full	Relevant	Highest ranked	First two ranked
Hamming-loss	0.20225	0.20272	0.22097	0.21114
Classification Accuracy	0.26404	0.27528	0.23596	0.25843
Precision	0.70133	0.69470	0.66002	0.68016
Recall	0.61376	0.61564	0.58474	0.60019
F-one	0.59785	0.59813	0.56648	0.58558
Accuracy	0.51695	0.51779	0.48619	0.50726
One-error	0.27528	0.27809	0.31461	0.29775
Coverage	0.59036	0.59223	0.67509	0.64279
Ranking loss	0.16131	0.16436	0.18892	0.17746
Average Precision	0.79728	0.79890	0.77358	0.78197
Average number of groups	**16.1**			

Table 4.12: Summary of results using ReliefF as relevance measure (XGBoost).

	Full	Relevant	Highest ranked	First two ranked
Hamming-loss	0.20037	0.19944	0.20974	0.20599
Classification Accuracy	0.26966	0.27528	0.25562	0.26404
Precision	0.70872	0.70629	0.68527	0.69167
Recall	0.61798	0.61938	0.60206	0.60627
F-one	0.60140	0.60262	0.58539	0.58858
Accuracy	0.52154	0.52341	0.50492	0.51007
One-error	0.27528	0.27809	0.29775	0.28933
Coverage	0.59410	0.58614	0.63296	0.61985
Ranking loss	0.16292	0.16212	0.17687	0.17073
Average Precision	0.79908	0.79913	0.78485	0.79036
Average number of groups	**23.0**			

The proposed RPFS procedures, specifically the model that includes only the relevant features, perform well when compared to the model including the full feature set. As in Section 4.5.1, the boxplots for the four relevance measures are also included in Figures 4.18 to 4.21.

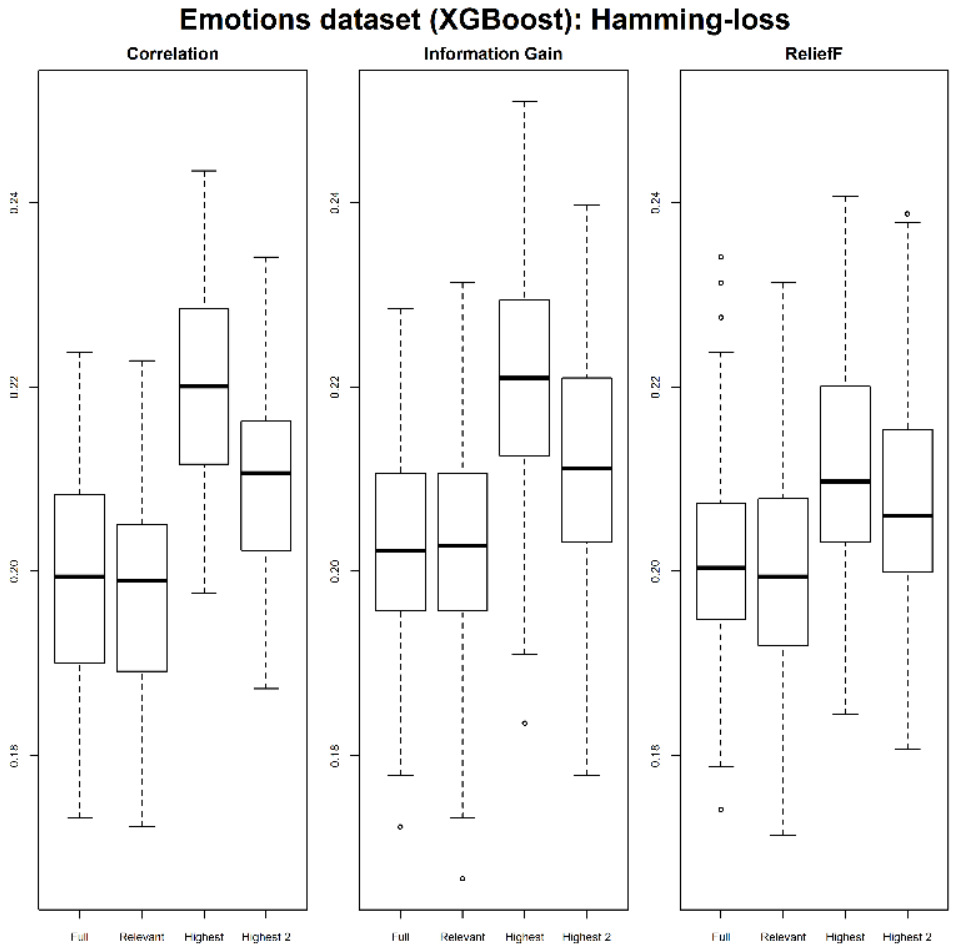


Figure 4.18 Comparison of relevance measures (XGBoost): Hamming-loss.

For Hamming-loss using the correlation coefficient or ReliefF as relevance measure, the medians for the FS procedure using only the relevant features are slightly smaller than the medians for the full models. The medians for the FS procedures that include fewer features are larger, but the results are still fairly similar, even though the procedures use fewer features. The boxplots in Figure 4.18 show that the variations in the results of the nine FS procedures are fairly similar.

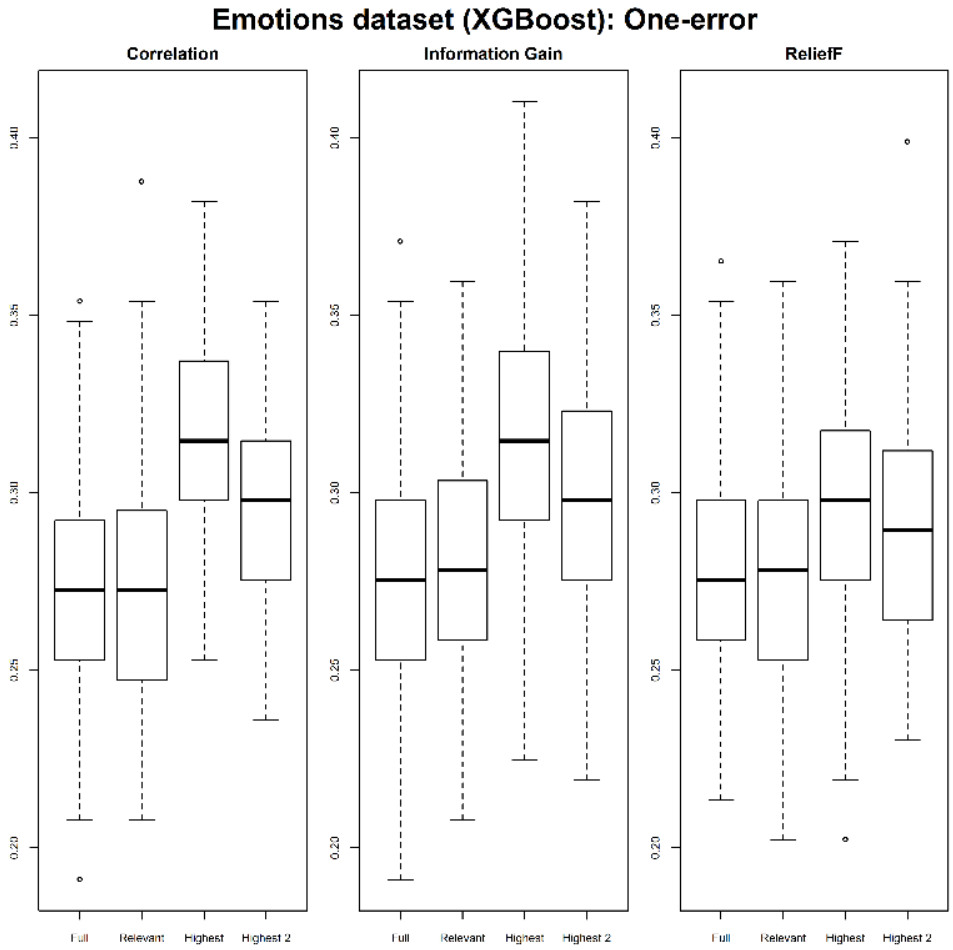


Figure 4.19 Comparison of relevance measures (XGBoost): One-error.

From the boxplots for One-error in Figure 4.19 one can conclude that the medians for the FS procedures using only the relevant features (and the correlation coefficient and ReliefF as relevance measures) are similar to those of the full model, but with slightly larger ranges and interquartile ranges.

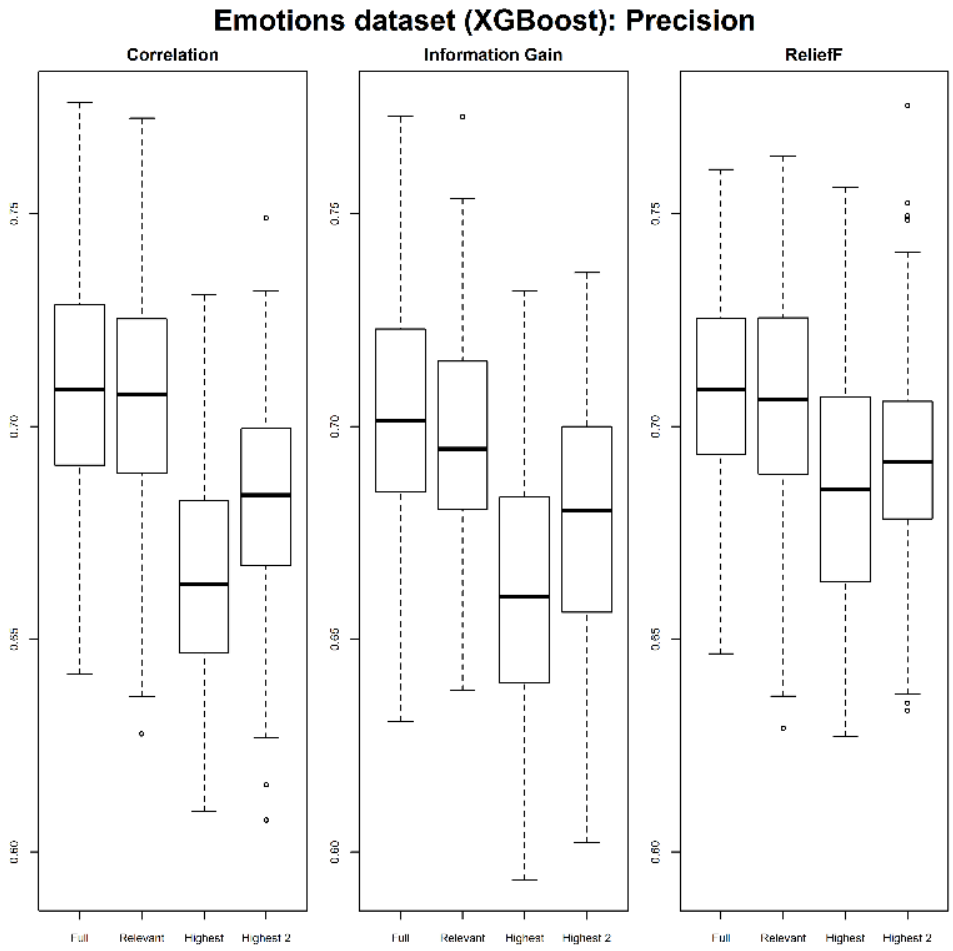


Figure 4.20 Comparison of relevance measures (XGBoost): Precision.

For Precision (the boxplots shown in Figure 4.20), the performances of the FS procedures using only the relevant features are similar to that of the full model, although the medians are slightly smaller. The FS procedures that include fewer features do not perform as well, but the results are still comparable. The variations in the values of Precision for the nine FS procedures are once again similar.

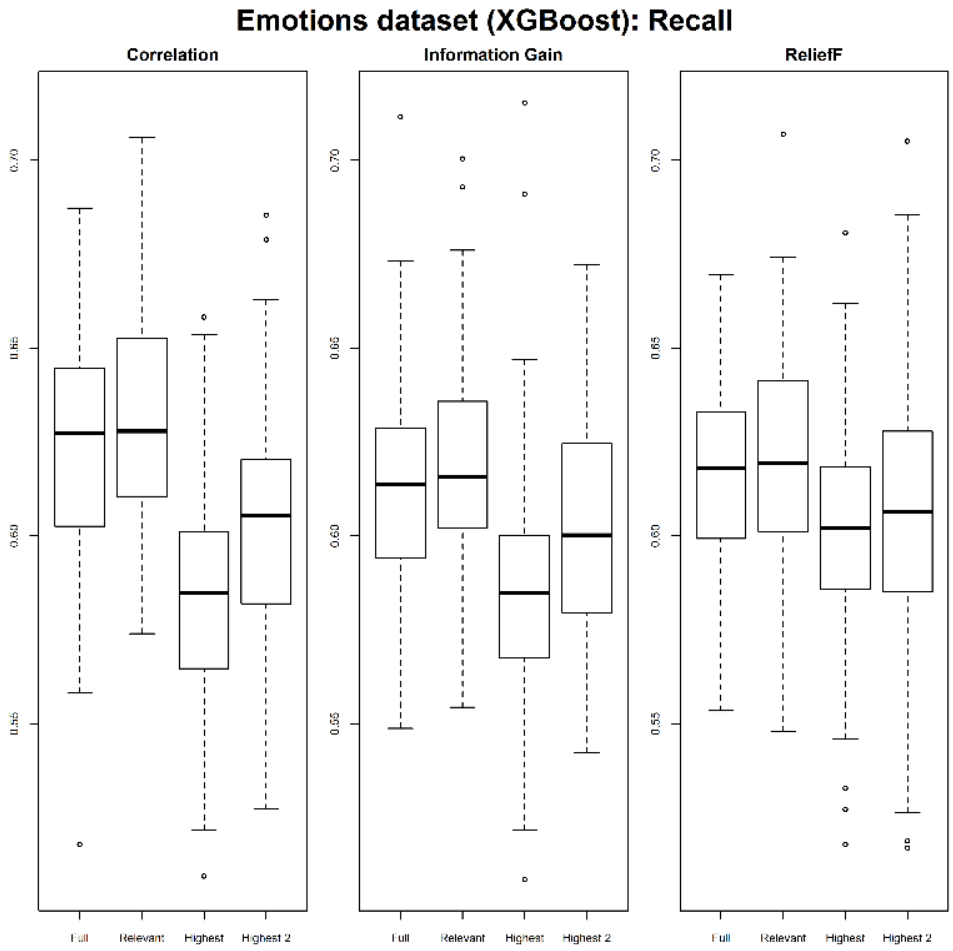


Figure 4.21 Comparison of relevance measures (XGBoost): Recall.

The performance in terms of Recall of the FS procedure using only the relevant features is similar to that of the full model. The FS procedures that include fewer features do not perform as well, but the results are still of the same order. The variations in the values of Recall for the nine FS procedures (refer to Figure 4.21) are once again similar.

In Figure 4.22, these boxplots are combined into a single graph in order to compare the different relevance measures for each of the four evaluation measures.

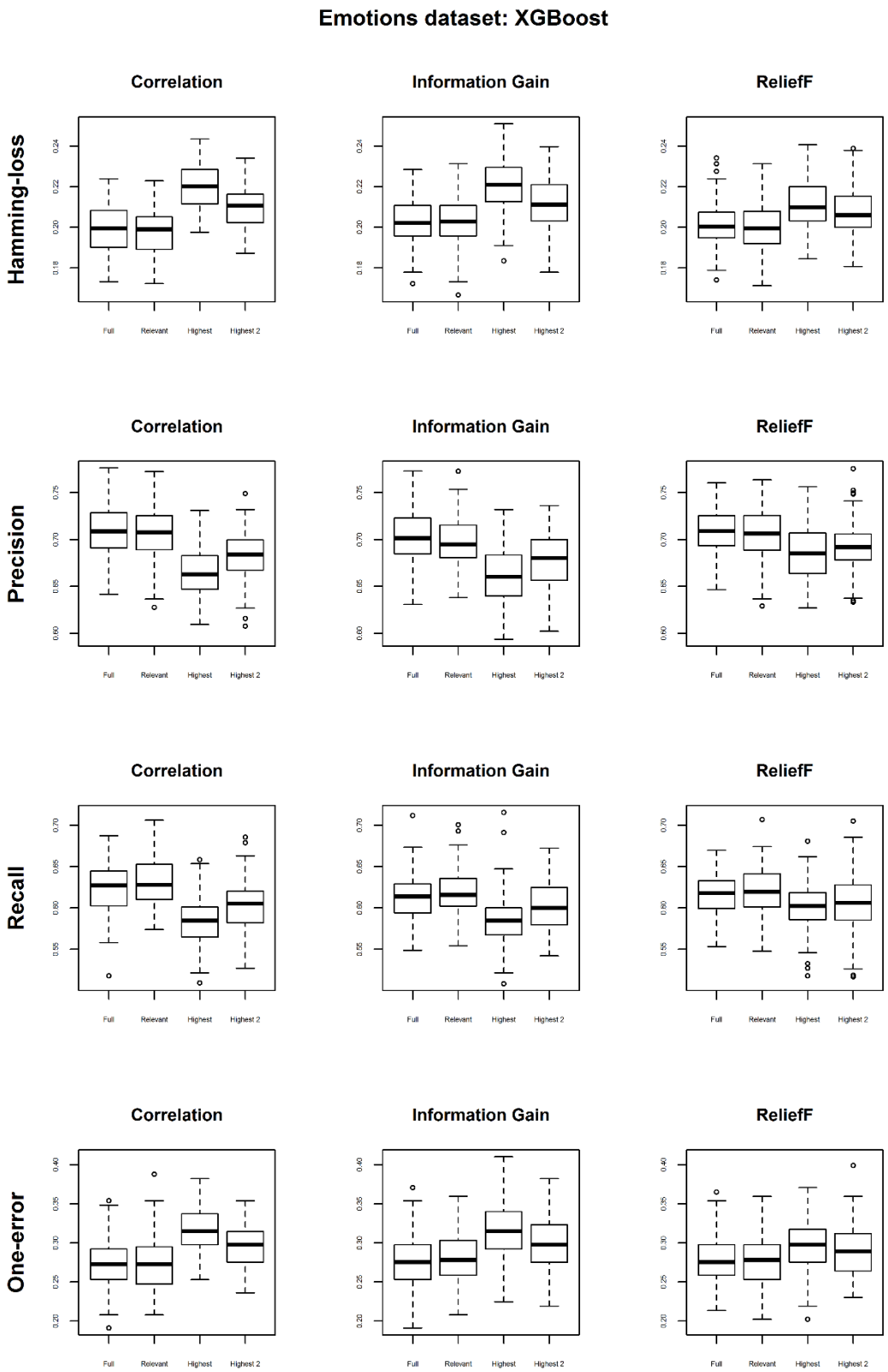


Figure 4.22 Comparison of relevance measures (XGBoost).

Again, if one only considers the results based on the correlation coefficient as relevance measure, the model that only selects the features which are deemed relevant performs only slightly better or worse than the full model. The model that only includes the highest ranked features consistently performs worse than the other models, including the full model. From the boxplots based on IG and ReliefF, similar conclusions can be drawn for each of the models. It should be noted that the model based on ReliefF which includes only the highest ranked feature performs better than its counterparts for the correlation coefficient and IG.

In Section 4.5.3, a comparison between the two classification algorithms will be performed.

4.5.3 Comparison between the SVM and XGBoost classifiers

In this section the classification algorithms, SVM and XGBoost, are compared. It is important to note that these classifiers do not play a role in the FS procedure. The focus is simply on the post-selection behaviour of these two approaches. The four evaluation measures, namely Hamming-loss, One-error, Precision, and Recall, are compared for the SVM and XGBoost classifiers. Boxplots are constructed for each of the nine FS procedures and these are plotted side by side in Figures 4.23 to 4.26.

The two classifiers are also compared based on the features that are considered irrelevant as well as the features that are ranked highest.

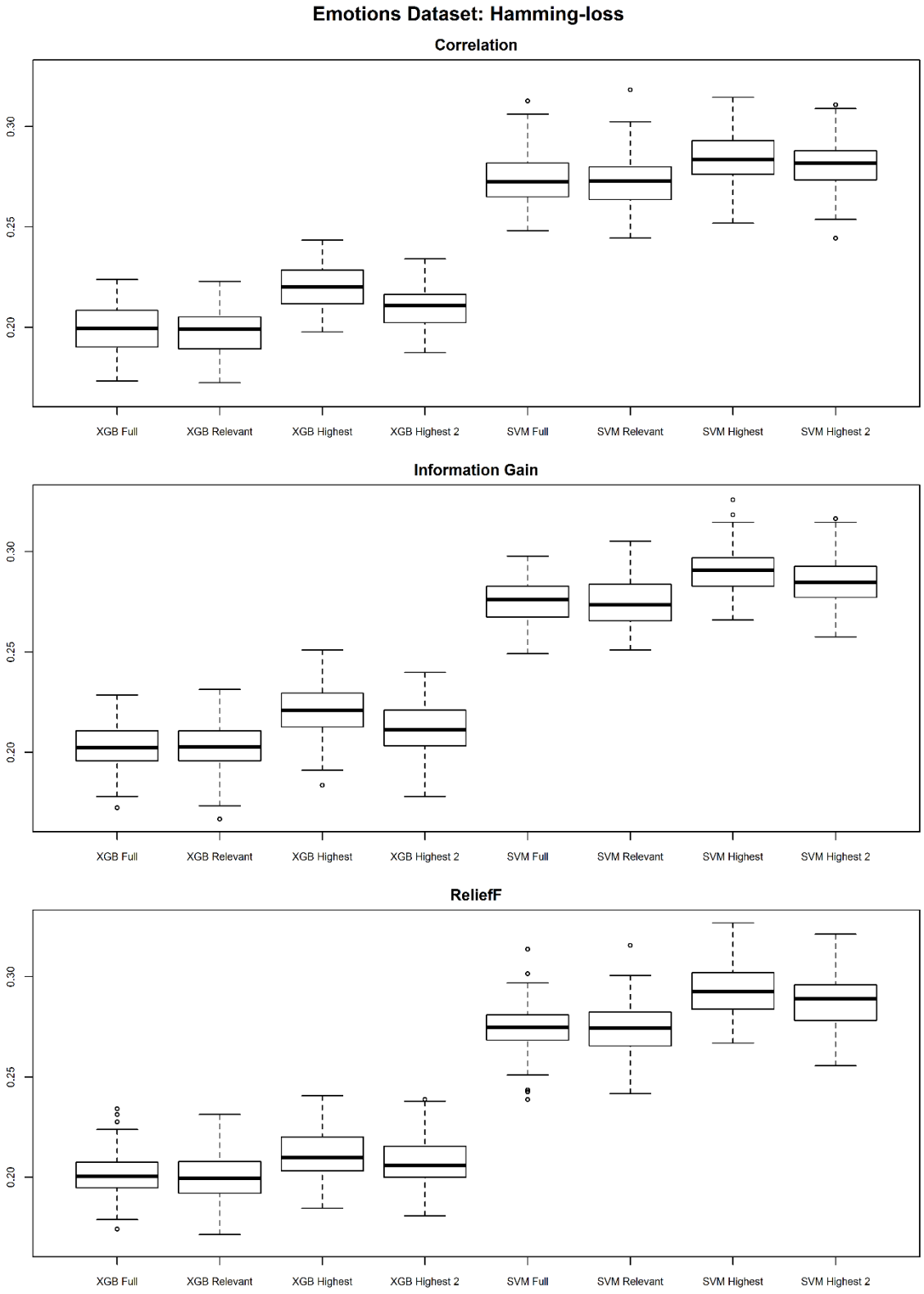


Figure 4.23 Comparison of SVM and XGBoost: Hamming-loss.

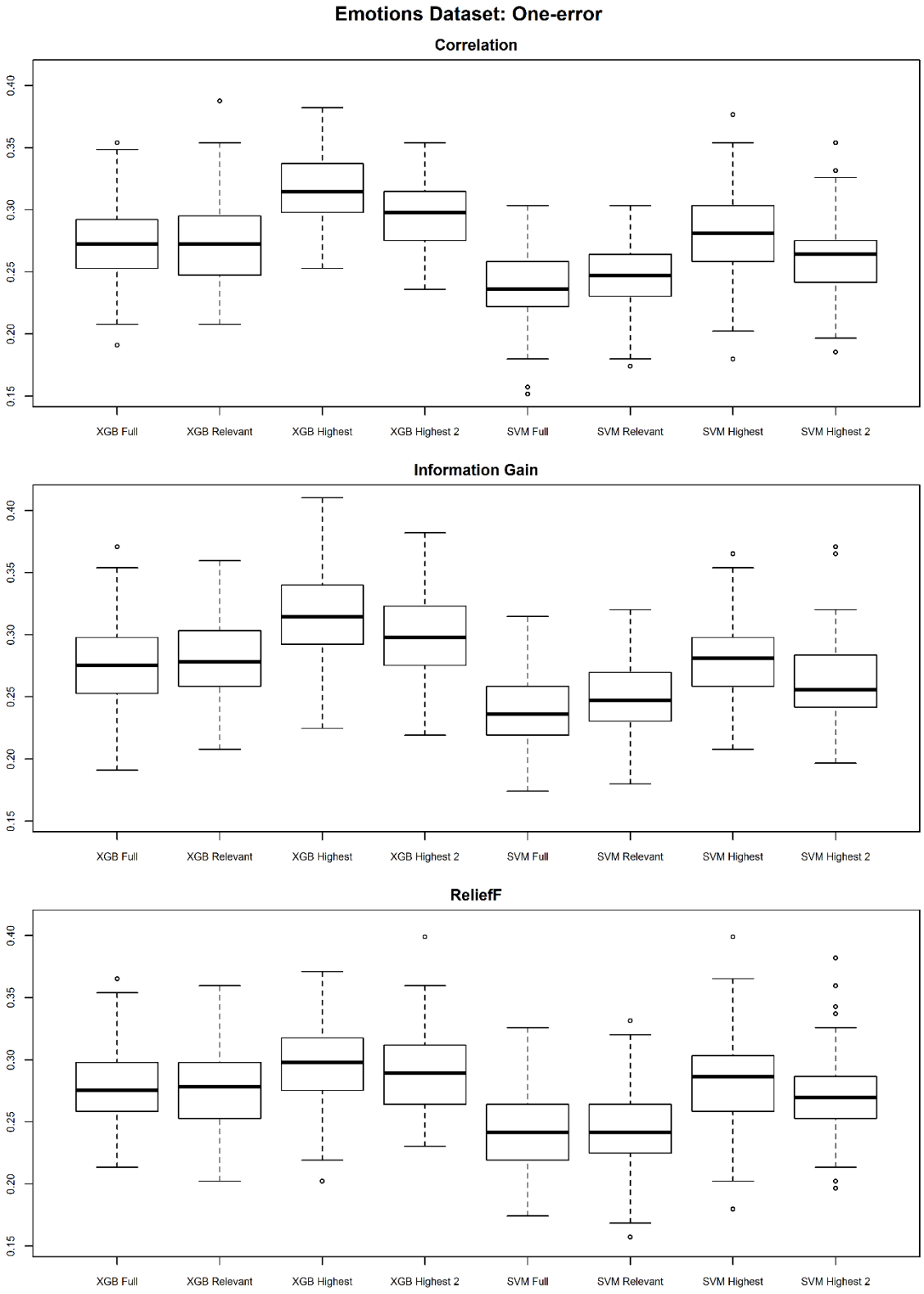


Figure 4.24 Comparison of SVM and XGBoost: One-error.

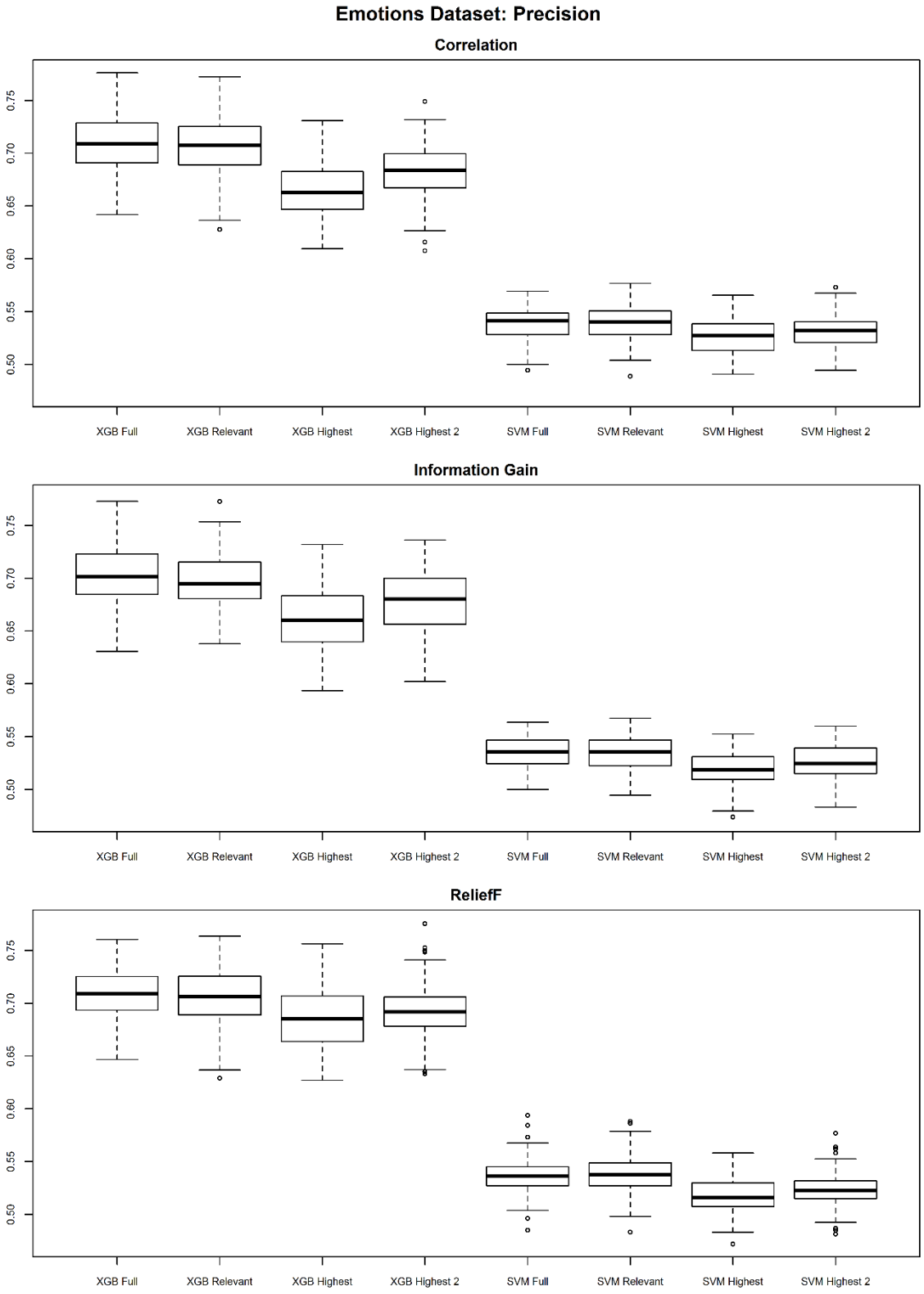


Figure 4.25 Comparison of SVM and XGBoost: Precision.

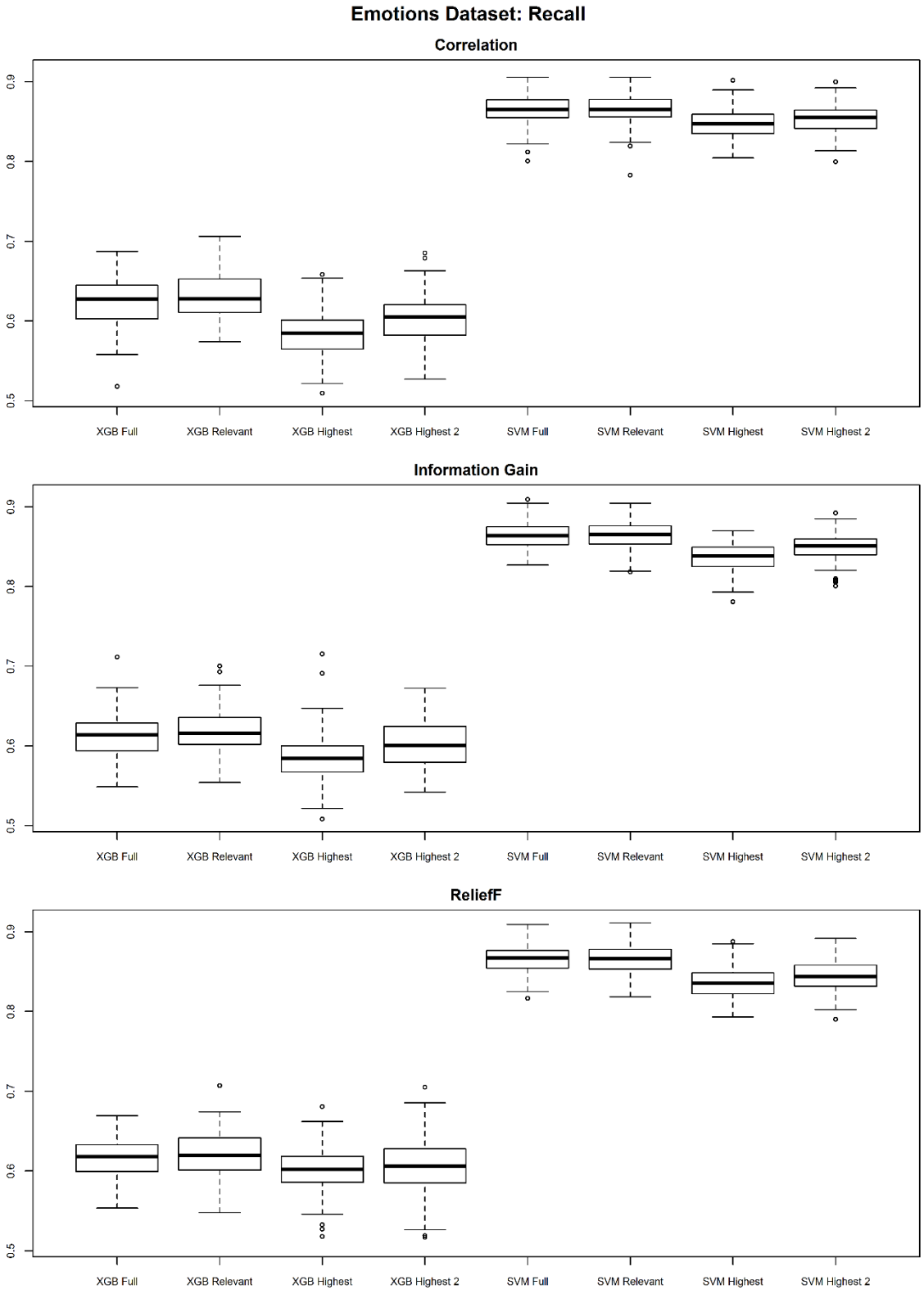


Figure 4.26 Comparison of SVM and XGBoost: Recall.

The XGBoost classifier performs better than the SVM for Hamming-loss across all three relevance measures in Figure 4.23. The medians of the evaluation measure for the XGBoost classifier are consistently lower than that for the SVM classifier. The variations associated with the Hamming-loss for the two classifiers are similar.

From Figure 4.24, one is able to see that the SVM classifier performs better than the XGBoost for One-error across all three relevance measures. The medians of the evaluation measure for the XGBoost classifier is consistently higher than that of the SVM classifier. The variations associated with the One-error for the two classifiers are similar.

For Precision, the XGBoost classifier consistently outperforms the SVM for all three relevance measures. However, there is less variations in the results for the SVM classifier – refer to Figure 4.25.

Finally, in Figure 4.26, the medians of the FS procedures using the SVM classifier are higher than those arising from the XGBoost classifier. Again, there seems to be less variation in the results for the SVM classifier.

In Figures 4.27 and 4.28 the number of times that a specific feature is considered irrelevant is plotted for each feature in a bar chart. For example, if one considers the FS procedure that uses the correlation coefficient as relevance measure and the SVM as classifier in Figure 4.27, feature 6 is considered to be irrelevant 57 times out of the 100 repetitions.

Two interesting observations can be made based on the results for the SVM classifier in Figure 4.27. The first relates to the relevance of the features relating to timbre (features 1 – 64) and those relating to rhythm (features 65 – 72)[2]. For all three relevance measures, in general, the timbre features are deemed irrelevant more often than the rhythmic features. The rhythmic feature *BH_HighLowRatio* (feature 69) is deemed to be irrelevant by all three relevance measures, but in less than 25% of the repetitions. The rhythmic features *BH_LowPeakBPM*, and *BH_ HighPeakBPM* are deemed to be irrelevant when using the correlation coefficient or IG, but not as frequently as the timbre features.

The second observation can be made with respect to the three relevance measures. For the correlation coefficient and for IG, a similar pattern can be observed when comparing the

[2] Refer to Appendix A for more information on these features.

selection frequencies. The only exception is feature 51 which is deemed to be irrelevant when using correlation, but not when using IG. The selection frequencies based on using ReliefF are different from the other two relevance measures, specifically when one considers features 20 – 34. For correlation and IG, features 20 – 34 are deemed to be irrelevant on some of the iterations, but this is not the case for ReliefF where these features are not deemed irrelevant even once.

Figure 4.28 allows one to draw a comparison based on the results for the XGBoost classifier. If the rhythmic features (features 65 – 72) are considered, one notices that the rhythmic features are deemed irrelevant less frequently than the timbre features. This result is consistent with the results obtained by the SVM classifier in Figure 4.27. All three relevance measures deem the rhythmic feature *BH_HighLowRatio* (feature 69) to be irrelevant, but in less than 30% of the repetitions. The rhythmic features *BH_LowPeakBPM* and *BH_ HighPeakBPM* are deemed to be irrelevant when using the correlation coefficient or IG, but not as frequently as the timbre features. This is also consistent with the results obtained in Figure 4.27.

As was the case for the results obtained using the SVM classifier, a similar pattern can be observed when comparing the selection frequencies for the correlation coefficient and IG. Feature 51, which is deemed to be irrelevant when using correlation again is the exception as it is not deemed irrelevant when using IG. Feature 51 is deemed irrelevant by ReliefF. When one compares the results from the SVM classifier with the XGBoost classifier, the results are similar for the correlation coefficient and for IG. The results between the SVM and XGBoost classifier based on ReliefF differ based on the frequency at which features 33 and 51 are deemed to be irrelevant. For SVM, feature 33 is deemed irrelevant only twice, but for XGBoost, it is deemed irrelevant 28 times. Feature 51 is deemed irrelevant only once for the SVM classifier but is deemed to be irrelevant 27 times for the XGBoost classifier.

The difference between the results based on the two classifiers can be attributed to the manner in which the thresholding was applied. The thresholds were optimised based on the SVM classifier, and then the same threshold was applied to the XGBoost classifier. This decision was made in order to be able to compare the results.

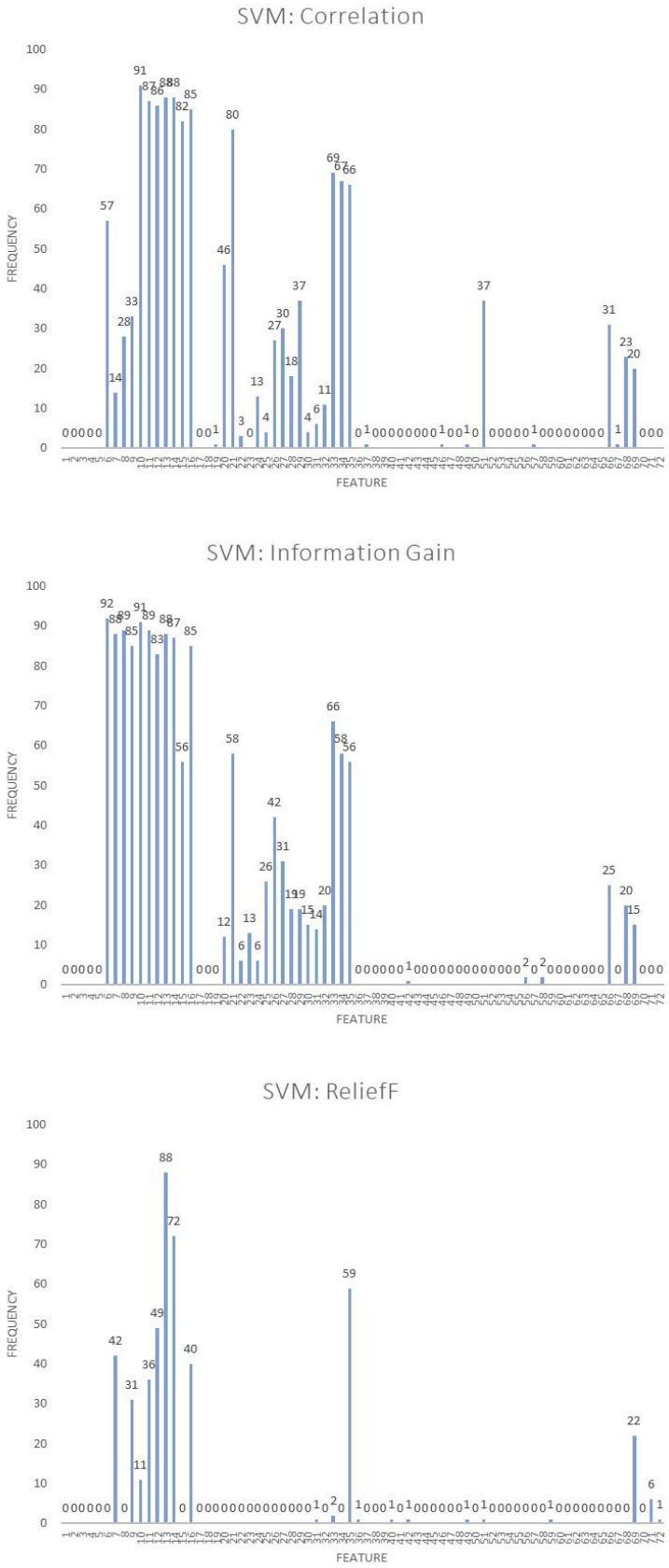


Figure 4.27 Comparison of selection frequencies for the irrelevant features (SVM).

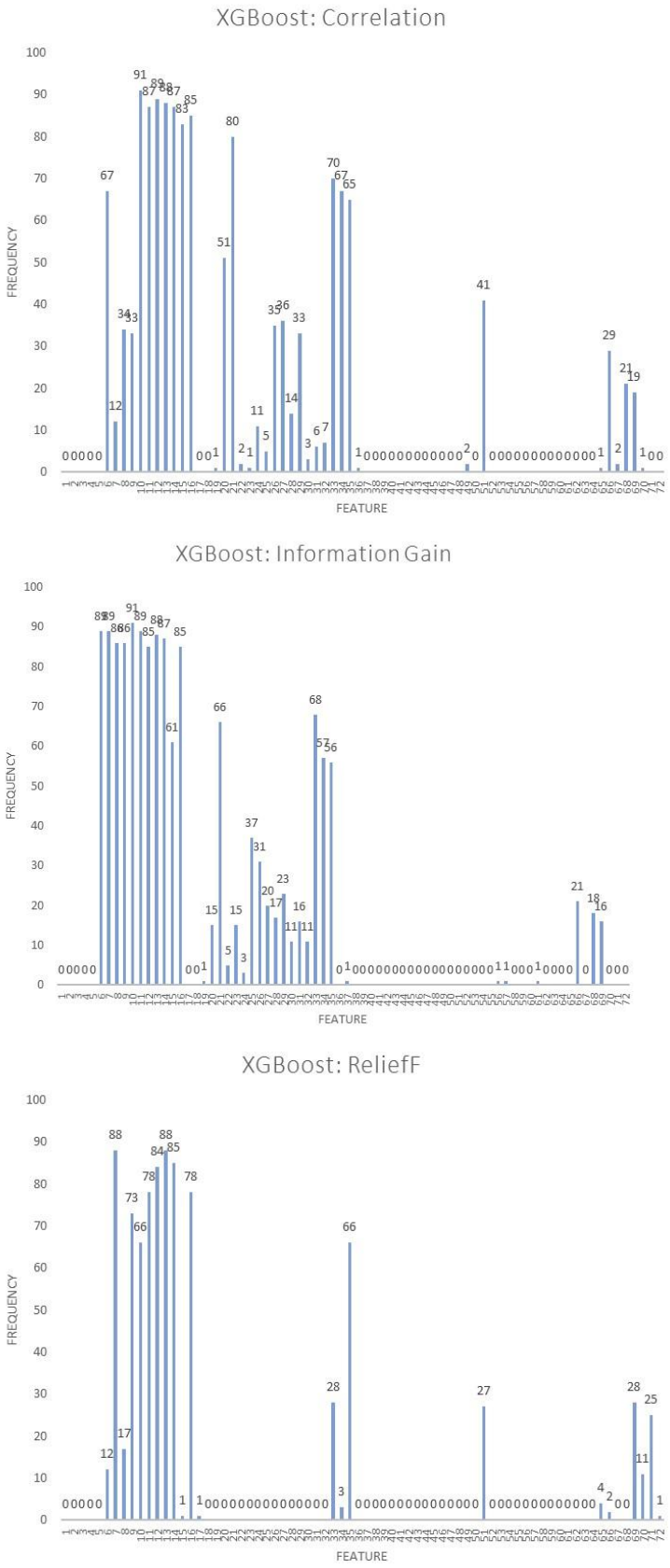


Figure 4.28 Comparison of selection frequencies for the irrelevant features (XGBoost).

In Figures 4.29 and Figure 4.30 the number of times that a specific feature is ranked as the most important feature by the procedure in its feature group is similarly represented in a bar plot. For example, when one considers Figure 4.29, and the FS procedure using ReliefF as relevance measure, one sees that feature 2 is ranked as the most important feature in its feature group 99 times out of the 100 iterations.

Some interesting observations that can be made from Figures 4.29 and 4.30 are:

1) For the correlation coefficient, for both the SVM and XGBoost classifier, features 3, 4, 5, 7, 8, and 18 are ranked highest in their feature group most frequently.
2) For IG, features 3 and 24 are ranked highest in their feature group most frequently for both the SVM and XGBoost classifier.
3) Feature 2 is ranked highest in its feature group 99 out of 100 repetitions for ReliefF, irrespective of the classifier.
4) The rhythmic features are ranked highest in their feature group less frequently than the features from the timbre category.

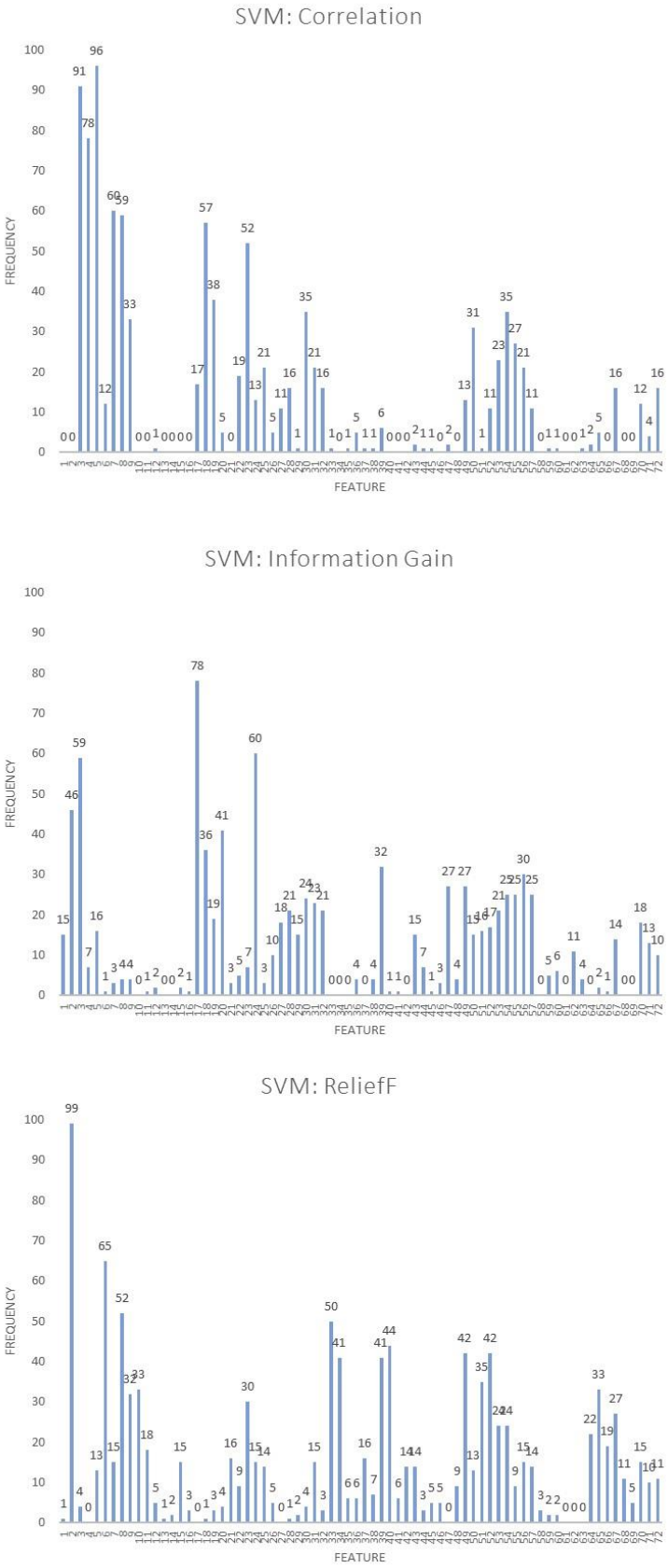


Figure 4.29 Comparison of selection frequencies for the highest ranked features (SVM).

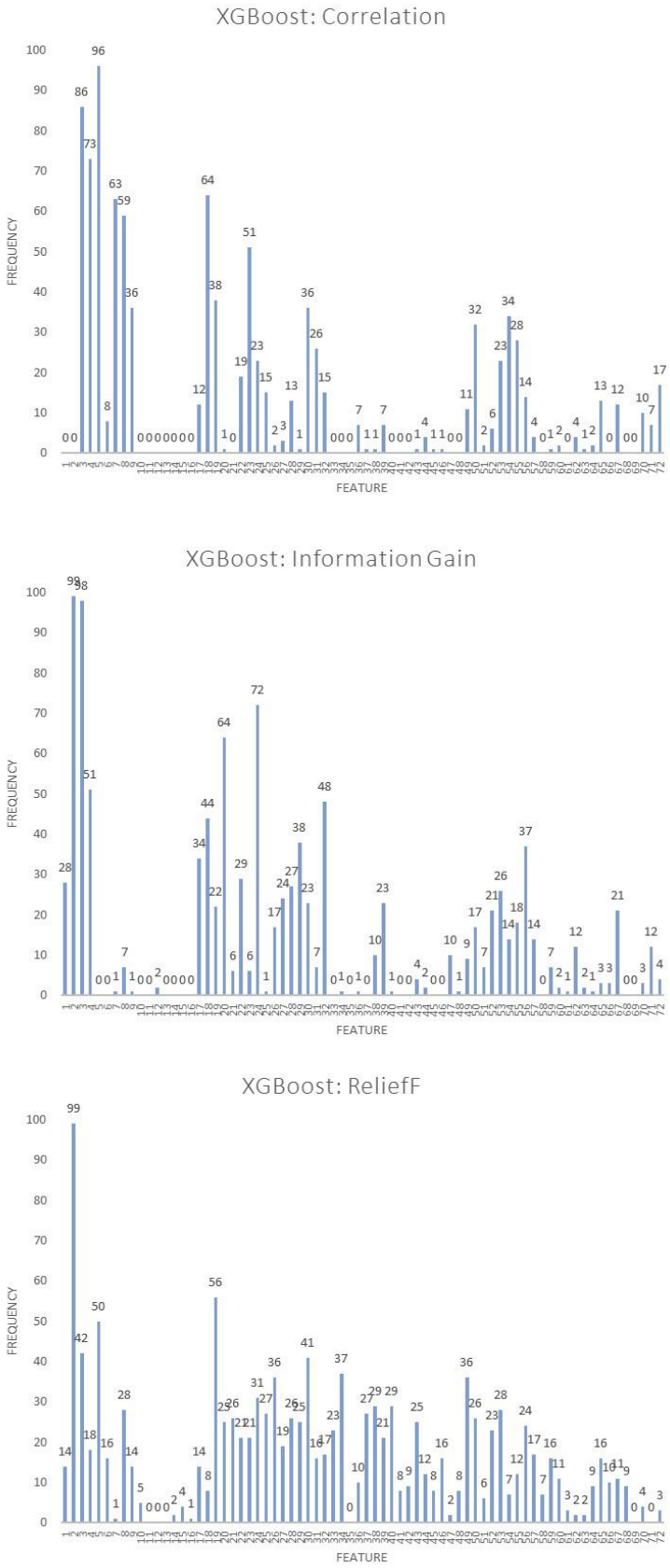


Figure 4.30 Comparison of selection frequencies for the highest ranked features (XGBoost).

In the following section, a concise summary of the results for the *Emotions* dataset will be given.

4.6 Conclusion

The chapter started with a short discussion on the multi-label benchmark datasets with specific attention being paid to the *Emotions* dataset. In Section 4.2, implementation of the proposed RPFS technique on the *Emotions* dataset was discussed. This section included detail on the construction of the relevance matrix with specific reference to the way in which the thresholds for relevance were determined. The MCA biplots and RPFS plots obtained for the *Emotions* dataset were shown as an example of the implementation of the proposed FS procedure. The specific choices regarding the problem transformation method and the base classifiers employed were also explained in this section.

The results of RPFS applying an SVM and an XGBoost classifier were then presented, followed by a comparison of the performance of these two classifiers in Section 4.5.

Four models were compared, namely the full model, the model that only includes the features that are deemed to be relevant features, the model that only includes the highest ranked feature from each feature group, and the model that includes the two highest ranked features from each feature group. The performance of the model that only includes the "relevant" features was found to be similar to that of the full model. The FS procedures that include fewer features did not perform as well, but the results are still fairly similar to those obtained from the full model, even though the procedures use substantially fewer features.

When comparing the three relevance measures, namely the correlation coefficient, IG, and ReliefF, it seems as if the results are similar when the SVM classifier is used. For the XGBoost classifier, there also does not seem to be much difference between the three relevance measures. It should, however, be noted that the model including only the highest ranked feature from each feature group identified by using ReliefF does perform better than the procedures based on the correlation coefficient and IG.

In the final section, the performance of the FS procedures based on the two classifiers were compared. For the evaluation measures Hamming-loss and Precision, the XGBoost classifier performs better than the SVM classifier, but the SVM classifier performs better than XGBoost

for Recall. The results for One-error were less clear, with the methods that utilise SVM as classifier performing only slightly better than those that use XGBoost as classifier.

In order to perform a comprehensive comparative study, synthetic datasets with more varied characteristics are required. The empirical investigation based on synthetic datasets is discussed in Chapter 5.

CHAPTER 5

EMPIRICAL INVESTIGATION: SYNTHETIC DATASETS

5.1 Introduction

While the benchmark datasets are useful for comparative studies of the performances of multi-label approaches, a need exists for synthetic datasets where the label cardinalities and densities are more varied. More specifically, when comparing and evaluating feature selection (FS) techniques, one would like to be able to generate synthetic data in such a manner that one is able to control the distribution of the features, the correlations amongst the labels, the label densities, as well as the local and global relevancies. This means that the user has knowledge about which features are important and which are irrelevant and/or redundant.

In this chapter, a brief discussion of synthetic multi-label data is provided. This is followed by an explanation of the experimental approach followed in this chapter. In the third section, the results of the proposed RPFS technique utilising MCA biplot methodology is discussed. In the final section, RPFS will be compared with the FS techniques proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013), as described in Section 3.4.

5.2 Synthetic multi-label data

In this section, some of the methods available for generating synthetic multi-label datasets will be presented. Secondly, the properties associated with multi-label datasets, specifically those relevant to generating synthetic datasets using the technique proposed by Sandrock and Steel (2015), are defined. Finally, a discussion of the cases considered in this dissertation is given which includes the properties of the 24 synthetic datasets that are used in the empirical investigation.

5.2.1 Methods for generating synthetic multi-label data

Limited research has been conducted with respect to the generation of synthetic multi-label data. One notable contribution by Read *et al.* (2012) develops a technique for generating multi-label data streams. Luaces *et al.* (2012) present a generator which searches for a hypothesis to

classify the inputs drawn from the uniform distribution and obtain a multi-label dataset with cardinality and density as close as possible to those specified by the user.

In Python's popular Scikit-learn a separate library `scikit-multilearn` has been developed specifically for multi-label classification. A random multi-label dataset can be generated using the function `make_multilabel_classification`. The function arguments allow the user to specify the label density (`n_labels`) and to allow for some unlabelled instances (`allow_unlabeled`). The function does not allow for a distinction between locally and globally relevant features.

Chou and Hsu (2005) extend a single-label proposal of Agrawal *et al.* (1992) to construct a synthetic multi-label dataset that uses specific functions to label instances. Noh *et al.* (2004) simulate datasets with different properties to compare multi-label decision trees.

Younes *et al.* (2011) use a synthetic dataset to investigate a new multi-label classifier. A covariance matrix and three labels are given, and instances are then labeled according to seven Gaussian distributions.

A study by Zhang *et al.* (2009) use hyperspheres to generate twelve artificial multi-label datasets to perform FS for naïve Bayes classification. *Mldatagen* is a web-based resource for generating artificial multi-label datasets based on hyperspheres and hypercubes developed by Tomás *et al.* (2014). Their proposal is based on the framework presented by Zhang *et al.* (2009).

Sandrock and Steel (2015) identify the following four shortcomings of the *Mldatagen* proposal:

1. Provision is not made for unlabelled instances.
2. No allowance is made for a specific multivariate distribution for the inputs. This implies that the user does not have control over the correlations between the features.
3. The correlations amongst the response variables, as well as the label densities cannot be controlled.
4. No distinction is made between local and global relevance.

Since the aim of this study is to perform multi-label FS, the second and fourth shortcomings are of critical importance and for this reason multi-label datasets are generated according to the method proposed by Sandrock and Steel (2015).

Two possible approaches to generating multi-label data mentioned by Sandrock and Steel (2015) are:

1) Generating $\mathbf{x}$ from its marginal distribution, followed by generating $\mathbf{y}$ from its conditional distribution given $\mathbf{x}$.
2) Generating $\mathbf{y}$ from its marginal distribution, followed by generating $\mathbf{x}$ from its conditional distribution given $\mathbf{y}$.

The method proposed by Sandrock and Steel (2015) focus on the second option. They consider the problem of generating a data instance $(\mathbf{x}'\mathbf{y}')'$ from an underlying distribution. The underlying distribution can be specified by the user.

Step 1: Generate $\mathbf{y}$

The suitable underlying distribution for $\mathbf{y}$ is a multivariate Bernoulli distribution. In its most general form, the specification of such a distribution requires that 2^q probabilities of the form $P(Y_1 = y_1,...,Y_q = y_q)$ are specified. For a large value of q, this is not feasible. Alternatively, if the labels can all be assumed to be independent, only probabilities of the form $P(Y_j = y_j)$, $j = 1,...,q$ need specification. This reduces the problem of simply generating q Bernoulli values. Sandrock and Steel (2015) advise that a popular approach in related literature is to specify $P(Y_j = y_j)$, $j = 1,...,q$ as well as $P(Y_i = y_i, Y_j = y_j)\ \forall\ i \neq j$. One can specify these joint probabilities by stipulating the correlations between the different variables. The authors denote $P(Y_j = 1)$ by p_j for $j = 1,...,q$.

Oman (2009) suggests four approaches for generating Bernoulli variables. Sandrock and Steel (2015) choose to apply the auto-regressive approach proposed by Oman (2009). Unfortunately, the realised correlations tend to differ from those specified by the user if the univariate probabilities are not equal. This problem is unavoidable and Sandrock and Steel (2015) comment that they are not aware of an approach that does not suffer from this downside.

The label vectors can be generated either conditionally or unconditionally. If the labels are generated conditionally, $\mathbf{y} = \mathbf{0}$ cases are discarded. These are kept when the label vectors are generated unconditionally. The terms *conditional* and *unconditional* are also used in a different

context in the discussion and Sandrock and Steel (2015) refer to the two different approaches of generating label vectors as *restricted* and *unrestricted*, respectively to avoid confusion.

Step 2: Generate $\mathbf{x}$ *from its conditional distribution given* $\mathbf{y}$.

Consider the case where the conditional distribution is a multi-variate normal distribution with mean vector dependent on $\mathbf{y}$ and constant covariance matrix. The mean vector needs to be specified in such a manner that the dependence on $\mathbf{y}$ is reflected. The proposal by Sandrock and Steel (2015) is as follows:

Specify a $p \times q$ binary matrix $\mathbf{A}$ with entries a_{ij}. The distribution of X_i is different if $Y_j = 1$ from the distribution when $Y_j = 0$. This dependence is modelled by taking $\mu_i(\mathbf{y}) = E[X_i \mid \mathbf{y}] = c\sum_{j=1}^{q} a_{ij} y_j$, $j = 1,...,q$.

If X_i is irrelevant for Y_j, *i.e.* $a_{ij} = 0$, $\mu_i(\mathbf{y})$ will not depend on y_j. However, if X_i is relevant for Y_j, *i.e.* $a_{ij} = 1$, $\mu_i(\mathbf{y})$ will depend on y_j, increasing by a positive quantity c if y_j changes from 0 to 1. The extent to which y_j changing from 0 to 1 influences $\mu_i(\mathbf{y})$ can be regulated by c. The value c could depend on i, this implies that a large value of c will correspond to a variable that is highly relevant for a given label. The value of c could be dependent on both i and j. This generalisation will reflect different degrees of relevance of X_i for different labels. Sandrock and Steel (2015) use a constant value of c.

The strength of the signal of $\mathbf{Y}$ for the input variables can be regulated by the quantity c. The signal strength is defined as $s^2 = \sum_{j=1}^{p} Var\left[E\left(X_j \mid \mathbf{Y}\right)\right]$. Also refer to Hastie *et al.* (2009: 649) for more information on the signal strength. If one considers a fixed value for i, according to the proposal outlined above, then

$$E[X_i \mid \mathbf{Y}] = c\sum_{j=1}^{q} a_{ij} Y_j \text{ and } Var[X_i \mid \mathbf{Y}] = c^2 E\left(\sum_{j=1}^{q} a_{ij} Y_j\right)^2 - c^2\left[E\left(\sum_{j=1}^{q} a_{ij} Y_j\right)\right]^2,$$

where $c^2\left[E\left(\sum_{j=1}^{q} a_{ij}Y_j\right)\right]^2 = c^2\left(\sum_{j=1}^{q} a_{ij}p_q\right)^2$. If one keeps in mind that Y_j is a binary variable, the first term can be expanded to $E\left(Y_jY_k\right) = \rho\left[p_j\left(1-p_j\right)p_k\left(1-p_k\right)\right]^{\frac{1}{2}} + p_jp_k$, the expression above becomes

$$Var\left[E\left(X_i \mid \mathbf{Y}\right)\right] = c^2\left\{\sum_{j=1}^{q} a_{ij}p_j + \rho\sum_{j=1}^{q}\sum_{k=1}^{q} a_{ij}a_{ik}\left[p_j\left(1-p_j\right)p_k\left(1-p_k\right)\right]^{\frac{1}{2}} + \sum_{j=1}^{q}\sum_{k=1}^{q} a_{ij}a_{ik}p_jp_k - \left(\sum_{j=1}^{q} a_{ij}p_j\right)^2\right\}$$

, for $j \neq k$ where ρ represents the constant correlation amongst all labels.

Therefore, the signal strength depends on c, $\mathbf{A}$, and ρ. Sandrock and Steel (2015) argue that it is undesirable for the signal to depend on the correlation amongst the labels. A proposal to specify c such that it is independent on ρ is made. They proceed to calculated c as

$$c = s \div \sum_{i=1}^{p}\left\{\sum_{j=1}^{q} a_{ij}p_j + \rho\sum_{j=1}^{q}\sum_{k=1}^{q} a_{ij}a_{ik}\left[p_j\left(1-p_j\right)p_k\left(1-p_k\right)\right]^{\frac{1}{2}} + \sum_{j=1}^{q}\sum_{k=1}^{q} a_{ij}a_{ik}p_jp_k - \left(\sum_{j=1}^{q} a_{ij}p_j\right)^2\right\}^{\frac{1}{2}}$$

, for $j \neq k$. This allows the user to specify s^2 to quantify the overall strength of the dependence between the features and the labels.

The results presented by Sandrock and Steel (2015) show that this specification of c does not completely attain the specified aim. The multi-label data generated does reflect the specified relevancies of the features with respect to the labels. However, it was found that correlation amongst the labels still influences the relationship between the features and the labels. The authors note that the proposed method does succeed in producing multi-label datasets that meet the required specifications.

In the following section, the relevant notation and properties of the synthetic datasets generated by using the method proposed by Sandrock and Steel (2015) will be discussed.

5.2.2 Properties of synthetic multi-label datasets

To discuss the synthetic multi-label datasets produced by using the technique proposed by Sandrock and Steel (2015), the following notation is introduced: the number of training instances, N, the number of irrelevant features, k, the number of relevant features, $p-k$, and the number of labels, q.

The resulting artificial dataset consists of an $N \times p$ matrix $\mathbf{X}$ of features and an $N \times q$ matrix $\mathbf{Y}$ of label responses. The parameter ρ is a value for the label correlation coefficient that allows the user to control the underlying label dependence. The procedure also allows the user to control the label densities as well as the strength of the signal. The label density of a dataset is controlled by specifying a vector of densities, D. For example, if a vector D equal to $\begin{bmatrix}0.5 & 0.4 & 0.3 & 0.2\end{bmatrix}$ is specified, it implies that label 1 will be present in 50% of the instances, label 2 in 40%, *etc*. The signal-to-noise ratio is user-specified and allows control over the overall strength of the dependence between the features and the labels. The signal strength is defined as $s^2 = \sum_{j=1}^{p} Var\left[E\left(X_j \mid \mathbf{Y}\right)\right]$ in the previous section.

To further control the relationship between the features and the labels, a $p \times q$ relevance matrix $\mathbf{A}$ is introduced. For illustration purposes, consider the following matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\\1\end{matrix}}\right\} p-k \\ \left.\vphantom{\begin{matrix}1\\1\end{matrix}}\right\} k \end{matrix}$$

Note that in this matrix, the rows correspond to the features and the columns to the labels. If $a_{ij} = 1$, it implies that feature i is (locally) relevant for label j. This means, for example, that for the above specification of the relevance matrix, feature 3 is relevant for labels 4 and 5, *etc*. The first three features are locally relevant, the fourth feature is globally relevant, and features 5 and 6 are irrelevant.

The relevance matrix, $\mathbf{A}$, is specified at the onset of the empirical investigation and is kept constant during simulation of the multi-label datasets. In the following section, the configurations of the 24 specific datasets used in this chapter will be presented.

5.2.3 Cases considered

The main aim of this dissertation is to investigate the performance of the proposed procedure on datasets with varied characteristics. In this section, the scope of this dissertation with respect to the cases considered is defined.

For all synthetic datasets, the number of relevant features, $p-k$, is constant at ten. To investigate the influence of noise, *i.e.* an increase in the number of irrelevant features, k, on the FS procedures, two scenarios are considered. In the first scenario, $k=p-k=10$, and in the second scenario, $k=5(p-k)=50$.

The number of training instances used in each dataset is dependent on the ratio of irrelevant features to relevant features. For the empirical study two ratios are applied, namely $0.5p$ and $4p$. The three levels for the number of training instances used are 30, 80, and 240. For $N=0.5(20)=10$, the SVM classifier often encounters an error if the dataset is generated in such a way that a label does not occur once in the training set. For this reason, the option $N=10$ is omitted from this study. Ten thousand testing instances are used for each of the 24 datasets. These 10 000 testing instances are generated using the same attributes as for the training data.

For each of the 24 datasets, the number of labels, q, is six. The parameter ρ, which allows the user to control the underlying label dependence, is investigated at two levels, namely 0 and 0.4. If the labels are assumed to be dependent, the correlation remains constant across all cases. Two different vectors of densities are used. For the first, the vector values are all specified to be fixed at a value of 0.4. This implies that each label will be present in approximately 40% of the instances. For the cases where values of the vector of densities are varied, vector D is set equal to $[0.25 \quad 0.31 \quad 0.20 \quad 0.42 \quad 0.28 \quad 0.35]$. It is important to note that initially these values were generated at random, but the procedure based on the SVM proved to be very sensitive to values smaller than 0.2 for the smaller datasets, where $N=30$. XGBoost is able to handle much smaller values for the densities. For example, the procedure was tested on a

scenario where a label was present for less than 5% of the instances. This is of particular importance when considering the problem of unbalanced data in (binary) classification.

The performance of the procedures is tested at two signal levels: for a weak signal, a signal-to-noise ratio of ten is used; and for a strong signal, a signal-to-noise ratio of 100 is used. A $p \times q$ relevance matrix, $\mathbf{A}$, is specified by the user to generate the synthetic datasets. It is important to note that this matrix is used to generate the training data, and that it differs from the empirical relevance matrix obtained using, for example, the correlation coefficient. In the generation of synthetic data in this study, the following $(p-k)\times q$ matrix was used. This remains fixed when constructing $\mathbf{A}$ for each dataset. The matrix used to define the relevance of each of the features for the different labels is

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Features 1 to 7, 9, and 10 are all locally relevant, while feature 8 is globally relevant. The procedure adds a row of zeroes to this matrix for every irrelevant feature assumed to be present. Table 5.1 provides a summary of the cases considered in the empirical study.

Table 5.1: Cases considered: 24 synthetic datasets.

	k	$p - k$	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 2	10	10	6	0.4	10	0.4	80	10 000
Dataset 3	10	10	6	0	100	0.4	80	10 000
Dataset 4	10	10	6	0.4	100	0.4	80	10 000
Dataset 5	10	10	6	0	10	vary	80	10 000
Dataset 6	10	10	6	0.4	10	vary	80	10 000
Dataset 7	10	10	6	0	100	vary	80	10 000
Dataset 8	10	10	6	0.4	100	vary	80	10 000
Dataset 9	50	10	6	0	10	0.4	240	10 000
Dataset 10	50	10	6	0	10	0.4	30	10 000
Dataset 11	50	10	6	0.4	10	0.4	240	10 000
Dataset 12	50	10	6	0.4	10	0.4	30	10 000
Dataset 13	50	10	6	0	100	0.4	240	10 000
Dataset 14	50	10	6	0	100	0.4	30	10 000
Dataset 15	50	10	6	0.4	100	0.4	240	10 000
Dataset 16	50	10	6	0.4	100	0.4	30	10 000
Dataset 17	50	10	6	0	10	vary	240	10 000
Dataset 18	50	10	6	0	10	vary	30	10 000
Dataset 19	50	10	6	0.4	10	vary	240	10 000
Dataset 20	50	10	6	0.4	10	vary	30	10 000
Dataset 21	50	10	6	0	100	vary	240	10 000
Dataset 22	50	10	6	0	100	vary	30	10 000
Dataset 23	50	10	6	0.4	100	vary	240	10 000
Dataset 24	50	10	6	0.4	100	vary	30	10 000

The following section presents the empirical approach which was followed to obtain the results in Sections 5.4 and 5.5.

5.3 Experimental approach

There are two main objectives of the empirical study in this chapter. The first objective is to present the results of the proposed RPFS procedure using the SVM and XGBoost base classifiers. The two classifiers are compared in terms of their performance on the 24 synthetic datasets. These results are presented in Section 5.4. The second objective entails a comparison

of RPFS with the FS procedures using probe selection (PS) and the procedure proposed by Spolaôr *et al*. (2013). This comparison is done in Section 5.5.

An extensive empirical study was performed on the 24 datasets described in Table 5.1 to achieve the objectives mentioned above. For each dataset configuration a training set was generated along with a large test dataset containing 10 000 instances.

5.3.1 Constructing the relevance matrix

After the training data is generated, the three relevance measures are used to construct a relevance matrix representing the perceived associations between the features and the labels. The threshold values for each of these measures are determined empirically using the SVM classifier. These thresholds are then applied when fitting the XGBoost classifier as well, in order for the results to be comparable. The threshold values applied for each dataset are provided in Table 5.2.

For the empirical relevance matrix, entries above the threshold are deemed to be relevant. For example, for Dataset 1, a feature is considered relevant if the absolute correlation coefficient between a feature and a label exceeds 0.4. The threshold values for the absolute correlation coefficient are in the range of 0.3 to 0.5 and are not particularly sensitive with respect to the characteristics of the dataset. For Dataset 1, an IG value exceeding the threshold value of 0.15 was deemed to be relevant. The thresholds for IG are in the interval 0.1 to 0.3 and the choice of threshold is more sensitive with respect to the characteristics of the datasets.

As mentioned in Section 3.3.2 two parameters need to be specified for the procedure using ReliefF, namely the number of ReliefF iterations, m, and the significance level, α. The threshold $\tau = 1/\sqrt{\alpha m}$ is used to determine whether a feature is identified as relevant or not. The w-values obtained using ReliefF are compared to τ and a feature with a w-value larger than τ is identified to be relevant. The number of ReliefF repetitions is set to 10 000 (the default value for the procedure implemented in R) and α ranges from 0.005 to 0.05. For ReliefF, the choice of α (and therefore the threshold) is more sensitive with respect to the characteristics of the datasets than for the procedures relying on the correlation between features and labels.

Table 5.2: Threshold values determined using the SVM classifier.

	Thresholds		
	ReliefF	**Correlation coefficient**	**IG**
Dataset 1	0.05	0.4	0.15
Dataset 2	0.05	0.4	0.15
Dataset 3	0.025	0.4	0.15
Dataset 4	0.01	0.5	0.25
Dataset 5	0.025	0.3	0.1
Dataset 6	0.025	0.3	0.1
Dataset 7	0.01	0.3	0.2
Dataset 8	0.005	0.4	0.25
Dataset 9	0.05	0.3	0.1
Dataset 10	0.05	0.4	0.2
Dataset 11	0.05	0.4	0.1
Dataset 12	0.01	0.4	0.25
Dataset 13	0.05	0.4	0.1
Dataset 14	0.01	0.5	0.3
Dataset 15	0.005	0.5	0.2
Dataset 16	0.01	0.5	0.3
Dataset 17	0.025	0.4	0.1
Dataset 18	0.05	0.4	0.25
Dataset 19	0.025	0.4	0.1
Dataset 20	0.05	0.4	0.25
Dataset 21	0.01	0.4	0.15
Dataset 22	0.01	0.5	0.3
Dataset 23	0.01	0.4	0.25
Dataset 24	0.01	0.5	0.3

For the procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013) – henceforth referred to as *PS* (for Probe Selection) and *Spolaôr* – these threshold values are determined empirically by the procedures and are not necessarily comparable to those in Table 5.2.

5.3.2 Performing feature selection using relevance pattern feature selection

As described previously, an MCA is performed on the relevance matrix to identify which features are irrelevant (*i.e.* which features are not considered relevant for any of the labels) and which features can be grouped together because they provide similar information.

In the exact same manner as in Chapter 4, the R libraries `UBbipl` and `UBFigs` (Le Roux and Lubbe, 2013) are employed to perform the MCA. An MCA biplot is constructed which provides a graphical representation of the different groups of features (feature groups). The column points obtained from the MCA are then removed and the row points are plotted in a RPFS plot. As in Section 4.4.2, four different models are proposed based on the RPFS plot. These four models are revisited here.

1. *Full*: All p features are included, *i.e.* no FS is performed.
2. *Relevant*: Remove the group of features that are deemed to be irrelevant. Let $\hat{k}$ = the estimated number of irrelevant features. Select only the $p - \hat{k}$ "relevant" features for the classification.
3. *Highest*: Select only the top ranked feature from each of the relevant feature groups. Features are ranked according to the values obtained from the relevance measures, *i.e.* the IG values, the absolute correlation coefficients, or the w-values for ReliefF.
4. *Highest 2*: Select only the top two ranked features from each of the relevant feature groups.

The classification approach applied in this chapter is discussed in the next section.

5.3.3 Classification

The classification approach used for the empirical investigation of the synthetic datasets in Chapter 5 is identical to the procedure applied to the *Emotions* benchmark dataset in Chapter 4. A given generated synthetic multi-label training dataset is transformed to single-label datasets in accordance with the BR approach. The single-label classifiers SVM and XGBoost are then applied to these transformed datasets, and the resulting classifiers are applied to the test data cases. The predicted labels are compared to the true labels of the test dataset and the four multi-label evaluation measures, Hamming-loss, One-error, Recall, and Precision are calculated.

One hundred synthetic training and test datasets are generated and the mean and median for each of the evaluation measures are calculated over the 100 repetitions. Due to the presence of outliers for some techniques, the results are also presented in boxplots. These representations allow for the comparison of methods based on location and variation. For the

detail on the specific algorithms used, refer to Section 4.4.3. The results will be discussed in the next section.

5.4 Results and conclusions

In this section the results using the two different classification techniques, namely SVM and XGBoost, are presented in Sections 5.4.1 and 5.4.2 respectively. The results for Dataset 1 are discussed in detail, but due to the extensive nature of the results, the other cases are dealt with in the appendices. A summary of all the results is presented in the form of a table providing the results of the Method of Pairwise Comparisons to rank the techniques across all 24 synthetic datasets.

Different groupings of the datasets will also be compared. The datasets are grouped based on the properties that are of interest, namely the signal strength, the number of irrelevant features, the number of training instances, the label dependence, and the vector of label densities. These groupings enable one to determine the influence of these aspects on the performances of the RPFS procedures. Finally, the number of features identified by each FS procedure will also be compared.

In Section 5.4.3 the results of RPFS using the SVM and XGBoost classifiers are discussed. Dataset 24 is used as an example and a three-way Analysis of Variance (ANOVA) is performed to determine whether the differences between the two approaches are statistically significant.

The techniques are also compared based on the various characteristics of the multi-label datasets in Section 5.4.4. These characteristics are once again the signal strength, the number of irrelevant features, the label dependence, and the vector of label densities. This comparison is carried out using four-way ANOVAs to determine whether the differences between the RPFS procedures are significant. Finally, the number of features identified by the different FS approaches is also compared.

5.4.1 Relevance pattern feature selection – Support vector machine classifier

In this section, the performance of the RPFS procedures implementing the SVM classifier is evaluated. The results obtained for each of the FS procedures for each of the 100 simulation repetitions are summarised in boxplots in Figures 5.1 to 5.4. From these boxplots, it is clear that a number of outliers occur for most of the FS procedures. For this reason, the medians and

interquartile ranges (IQRs), instead of the means and standard deviations, are used for the remainder of the comparisons.

As an illustrative example, the results for Dataset 1 with the following properties will be discussed: number of irrelevant features, $k = 10$; number of relevant features, $p - k = 10$; number of labels, $q = 6$; no dependence among the labels, $\rho = 0$; signal strength, 10; each label will be present in approximately 40% of the instances; number of training instances, 80; and number of test instances, 10 000.

In Figures 5.1 to 5.4 the performance for each of the three relevance measures across each of the four different evaluation measures for the SVM classifier are considered.

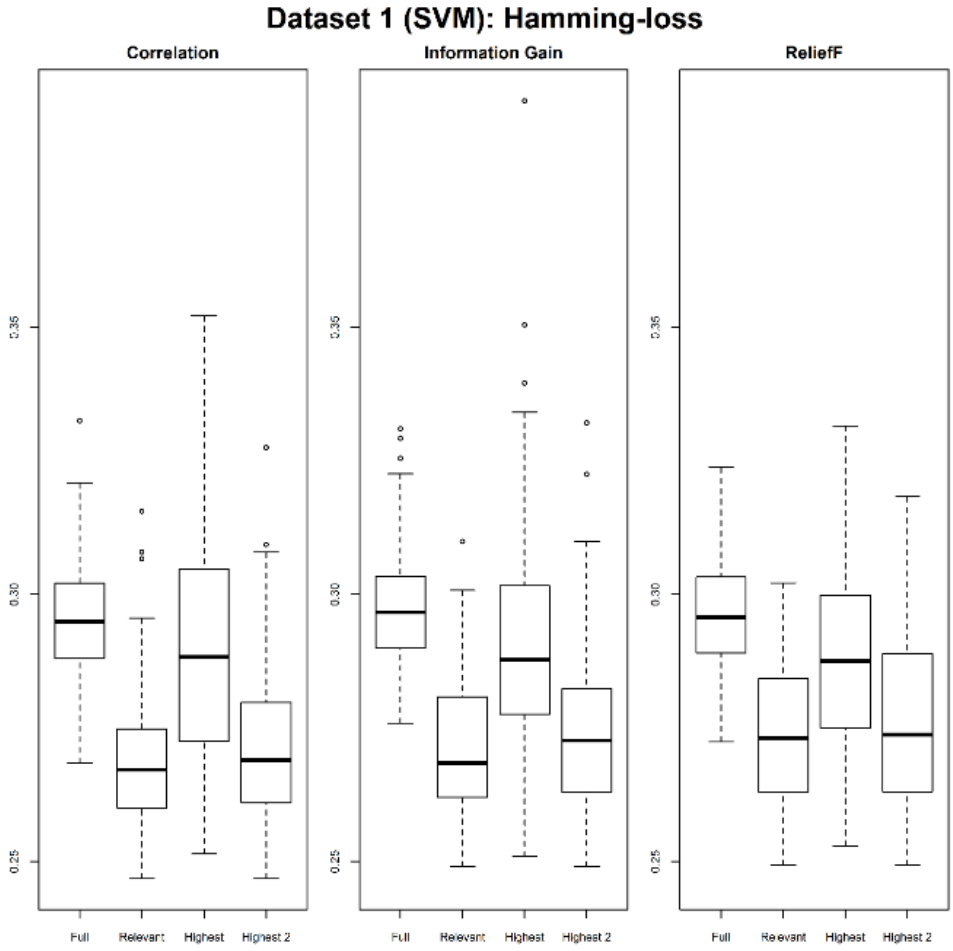


Figure 5.1 Comparison of Hamming-loss using the SVM classifier: Dataset 1.

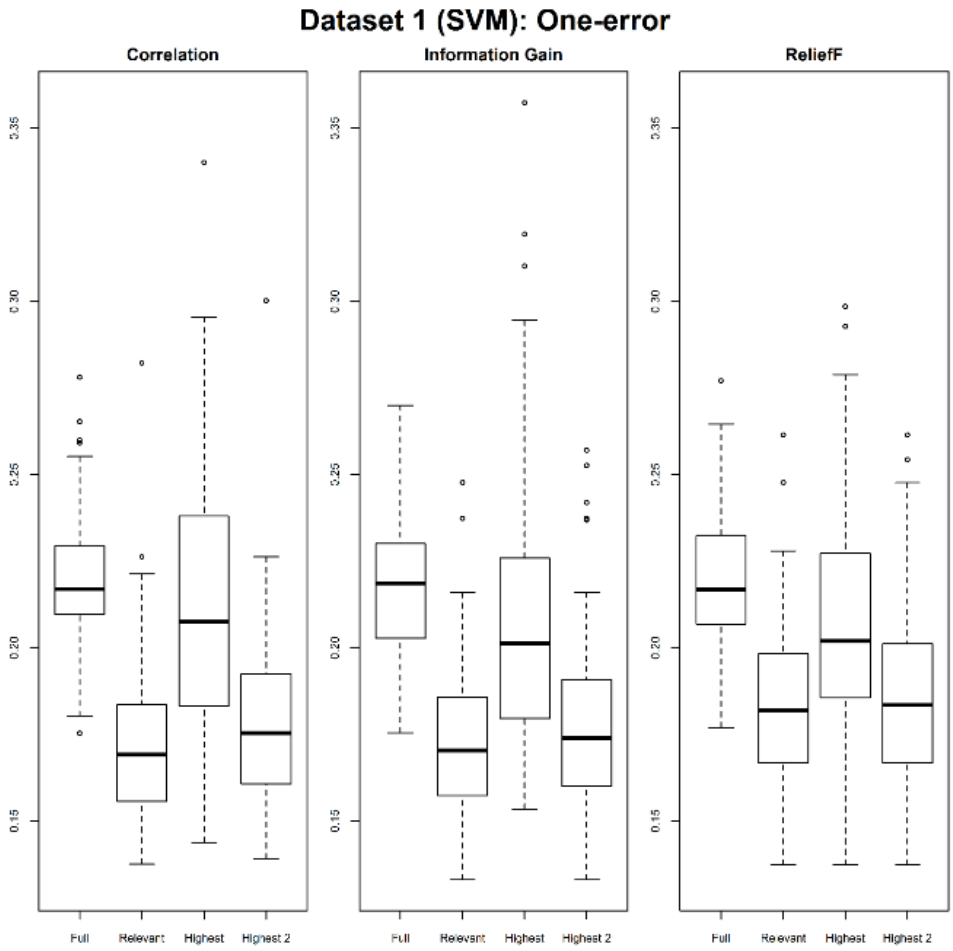


Figure 5.2 Comparison of One-error using the SVM classifier: Dataset 1.

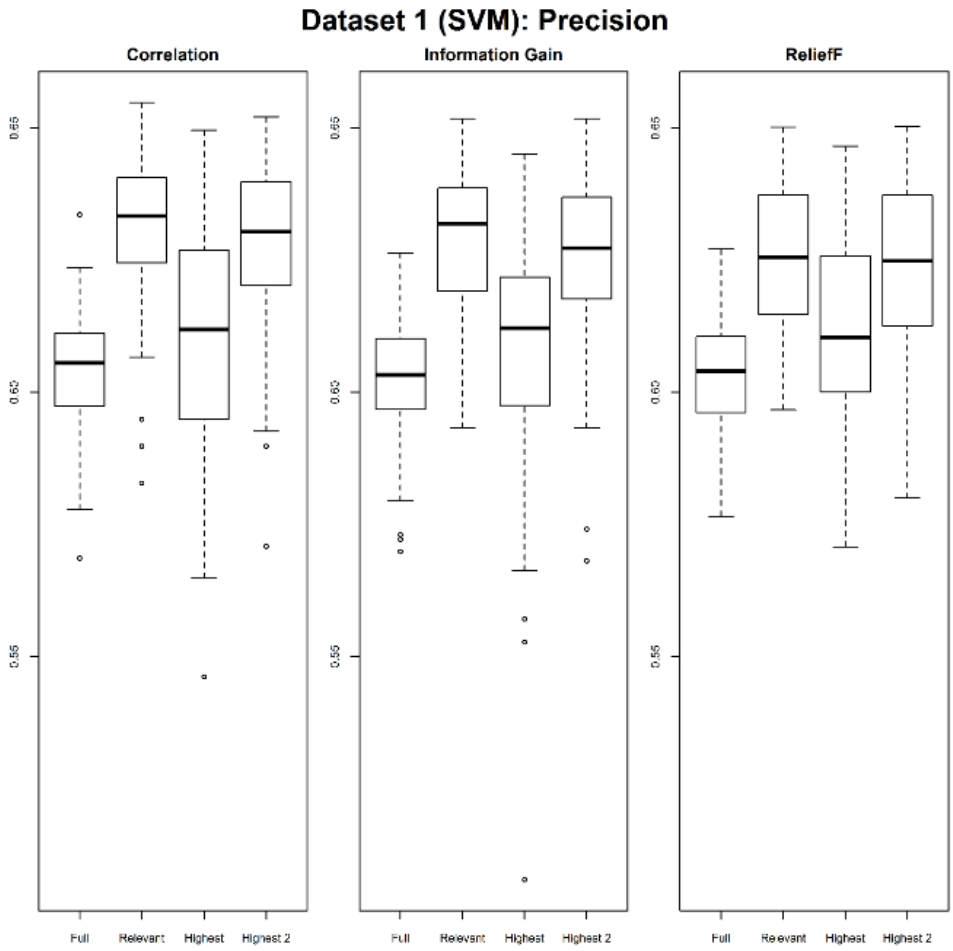


Figure 5.3 Comparison of Precision using the SVM classifier: Dataset 1.

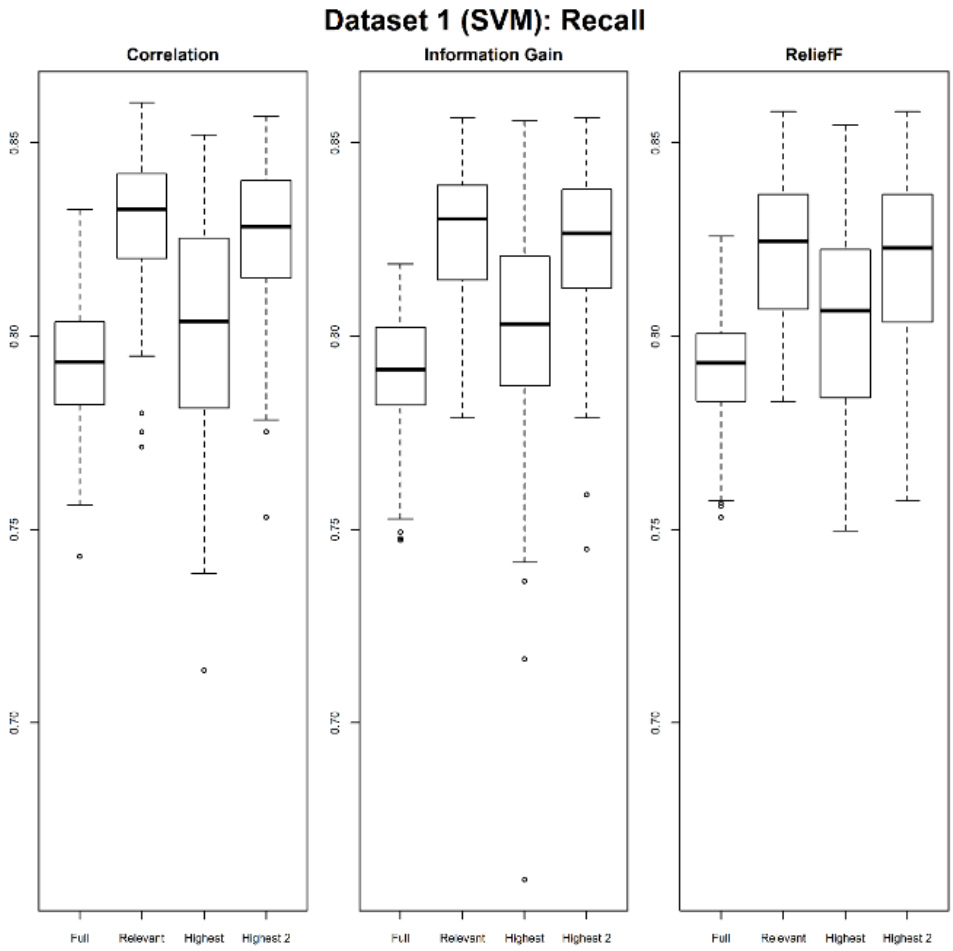


Figure 5.4 Comparison of Recall using the SVM classifier: Dataset 1.

For all three relevance measures (the correlation coefficient, IG, and ReliefF), across all four evaluation measures (Hamming-loss, One-error, Precision, and Recall) the reduced models perform better than the full model, with the procedure that includes all features which are deemed relevant, performing best. The procedure that includes the two highest ranked features from each group of relevant features performs relatively well compared to the Relevant procedure. There seems to be more variation in the procedures that only uses the highest ranked feature from every feature group.

Similar graphs have been generated for all 24 synthetic datasets. The rest of these graphs are available in Appendix B[3]. While it is useful to view each evaluation measure separately, it becomes difficult to compare the results across multiple datasets. Figure 5.5 provides a summary by combining Figures 5.1 to 5.4 in a single graph.

[3] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

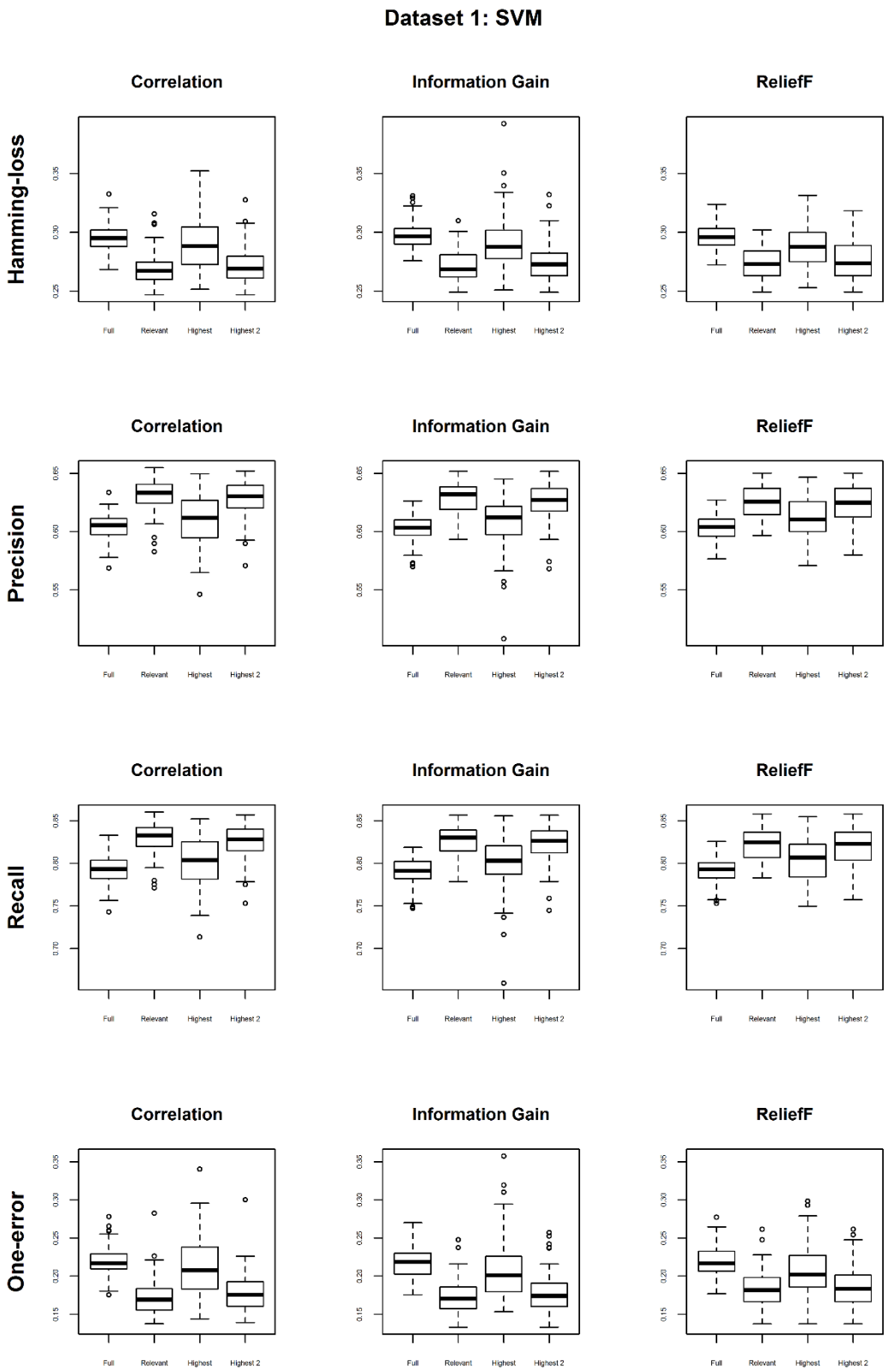


Figure 5.5 Summary of results for the SVM classifier: Dataset 1.

Figure 5.5 allows for the comparison of the three relevance measures in terms of each of the four evaluation measures. The results based on the correlation coefficient, IG, and ReliefF are similar. The medians and the IQRs for the three relevance measures are similar. It also allows for the comparison of different FS procedures. Consider for example the results based on the correlation coefficient as relevance measure. For all four evaluation measures all of the reduced models perform better than the full model. It can be seen that the model that only includes the highest ranked feature from each feature group has a much larger variation than the other models. A similar pattern with respect to the FS models can be observed for both IG and ReliefF.

These summaries are available for all 24 synthetic datasets. They are available in Appendix D[4]. This type of representation does not allow for the comparison of the results for all 24 datasets that were used in the empirical study. A ranking of the procedures which includes a measure of variation in performance is necessary in order to evaluate the merits of the different procedures. The Method of Pairwise Comparisons is used to compare ranked preferences in scientific studies and will provide us with the desired rankings.

For the Method of Pairwise Comparisons, each procedure is matched one-on-one with each of the other procedures. A procedure receives one point for a one-on-one win and half a point for a tie. The procedure with the highest total is declared the winner. The Method of Pairwise Comparisons uses all the information from the preferences but does not use all the information at once. This means that as pairs of procedures are matched up, any information available about the remaining procedures is ignored (Csima, 2014).

To apply the Method of Pairwise Comparisons, the median for each of the four evaluation measures for each of the procedures is calculated, and the procedures are ranked accordingly. An existing *Excel* macro[5] is adapted to apply the Method of Pairwise Comparisons to the results of this investigation.

The resulting rankings for each evaluation measure are presented in Tables 5.3 to 5.6. The IQRs are included to provide insight into the amount of variation that is present in the results. A darker green indicates a larger IQR and the full model is shaded in pink. The full model here

[4] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.
[5] *Canticum Novum* – Top 10 Excel Macro – T.L. Berning (2006).

is denoted as RPFS Full in all of the tables. The rationale for this will be explained in Section 5.5.

Table 5.3: Hamming-loss for SVM.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Highest	Cor Highest	RPFS Full
	IQR	0.0146	0.0187	0.0184	0.0192	0.0211	0.0258	0.0246	0.0239	0.0321	0.0138
	Feature Reduction	52	52			46		57	62	63	
Dataset 2	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0084	0.0102	0.0095	0.0132	0.0107	0.0170	0.0171	0.0178	0.0246	0.0089
	Feature Reduction	54	54			48	61		63	61	
Dataset 3	Median	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0052	0.0052	0.0058	0.0049	0.0049	0.0057	0.0060	0.0066	0.0098	0.0093
	Feature Reduction	50		51	50		51	48		52	
Dataset 4	Median	IG Relevant	ReliefF Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0042	0.0050	0.0056	0.0047	0.0071	0.0104	0.0079	0.0157	0.0214	0.0082
	Feature Reduction	50	50		50			57	60	65	
Dataset 5	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0096	0.0096	0.0089	0.0100	0.0104	0.0140	0.0145	0.0157	0.0180	0.0113
	Feature Reduction	48		51		44	49		36	53	
Dataset 6	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0087	0.0086	0.0116	0.0108	0.0098	0.0131	0.0166	0.0149	0.0168	0.0077
	Feature Reduction	49	45			41		61	58	57	
Dataset 7	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full	IG Relevant	IG Highest 2	IG Highest
	IQR	0.0061	0.0059	0.0057	0.0057	0.0061	0.0054	0.0126	0.0296	0.0290	0.0356
	Feature Reduction	49		48		51	48		67		77
Dataset 8	Median	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	ReliefF Highest 2	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0061	0.0060	0.0052	0.0076	0.0064	0.0065	0.0097	0.0085	0.0134	0.0062
	Feature Reduction	50		50	53	50			58	63	
Dataset 9	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0046	0.0046	0.0035	0.0035	0.0035	0.0045	0.0049	0.0104	0.0141	0.0090
	Feature Reduction	83		83	83		83		84	85	
Dataset 10	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	IG Highest 2	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0322	0.0329	0.0399	0.0331	0.0283	0.0279	0.0306	0.0360	0.0290	0.0303
	Feature Reduction	72		84	79		57		77	48	
Dataset 11	Median	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0040	0.0044	0.0038	0.0043	0.0106	0.0141	0.0156	0.0204	0.0198	0.0052
	Feature Reduction	84	84	84		86			88	89	
Dataset 12	Median	IG Highest	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0187	0.0186	0.0194	0.0163	0.0153	0.0163	0.0245	0.0255	0.0229	0.0094
	Feature Reduction	82		76	73		82	86		90	
Dataset 13	Median	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0368	0.0416	0.0347	0.0371	0.0343	0.0301	0.0312	0.0308	0.0508	0.0385
	Feature Reduction	83		83	83		83	83		87	
Dataset 14	Median	Cor Highest 2	Cor Relevant	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0308	0.0301	0.0312	0.0343	0.0347	0.0371	0.0416	0.0368	0.0508	0.0385
	Feature Reduction		81	84		80	86		83	89	
Dataset 15	Median	Cor Relevant	IG Relevant	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0027	0.0023	0.0027	0.0021	0.0027	0.0026	0.0054	0.0044	0.0047	0.0056
	Feature Reduction	83	83		83			86	86	86	
Dataset 16	Median	Cor Relevant	Cor Highest 2	IG Highest 2	IG Relevant	ReliefF Relevant	IG Highest	ReliefF Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0195	0.0252	0.0233	0.0209	0.0199	0.0316	0.0224	0.0289	0.0278	0.0179
	Feature Reduction	81			80	78	84		87	84	
Dataset 17	Median	Cor Relevant	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0043	0.0050	0.0051	0.0061	0.0060	0.0056	0.0106	0.0090	0.0132	0.0065
	Feature Reduction	84	83		83			85	84	86	
Dataset 18	Median	Cor Highest	Cor Highest 2	IG Highest 2	IG Relevant	Cor Relevant	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0310	0.0264	0.0290	0.0287	0.0305	0.0356	0.0272	0.0273	0.0259	0.0221
	Feature Reduction	83			77	72	88	73		38	
Dataset 19	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0034	0.0042	0.0036	0.0036	0.0047	0.0049	0.0065	0.0057	0.0096	0.0053
	Feature Reduction	84		85		85	83		85	86	
Dataset 20	Median	IG Relevant	IG Highest 2	IG Highest	Cor Highest 2	Cor Relevant	Cor Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0156	0.0174	0.0165	0.0160	0.0130	0.0188	0.0163	0.0150	0.0156	0.0149
	Feature Reduction	77		84		73	81	70		43	
Dataset 21	Median	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0028	0.0028	0.0028	0.0022	0.0022	0.0027	0.0027	0.0028	0.0022	0.0074
	Feature Reduction	83	83		83		83		84	84	
Dataset 22	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0197	0.0216	0.0260	0.0237	0.0235	0.0273	0.0281	0.0266	0.0259	0.0298
	Feature Reduction	81		85		81	86		85	78	
Dataset 23	Median	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0026	0.0026	0.0029	0.0026	0.0026	0.0037	0.0070	0.0162	0.0245	0.0042
	Feature Reduction	83		83	83	83		88		92	
Dataset 24	Median	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest	Cor Highest 2	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0207	0.0216	0.0160	0.0190	0.0201	0.0177	0.0230	0.0211	0.0227	0.0181
	Feature Reduction	80		81	84	84		82		75	

Table 5.4: One-error for SVM.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0276	0.0281	0.0303	0.0316	0.0314	0.0342	0.0459	0.0414	0.0541	0.0192
	Feature Reduction	52	52			46		62	57	63	
Dataset 2	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0189	0.0230	0.0200	0.0296	0.0224	0.0355	0.0321	0.0444	0.0418	0.0235
	Feature Reduction	54	54			48	61		63	61	
Dataset 3	Median	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0041	0.0044	0.0041	0.0037	0.0037	0.0038	0.0046	0.0043	0.0063	0.0151
	Feature Reduction	50	51		50		51	48		52	
Dataset 4	Median	IG Relevant	ReliefF Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0144	0.0132	0.0171	0.0183	0.0212	0.0292	0.0242	0.0370	0.0377	0.0229
	Feature Reduction	50	50		50			57	60	65	
Dataset 5	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0228	0.0220	0.0263	0.0215	0.0211	0.0269	0.0257	0.0298	0.0360	0.0259
	Feature Reduction	48		51		44	49		36	53	
Dataset 6	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0195	0.0214	0.0310	0.0308	0.0234	0.0277	0.0482	0.0360	0.0321	0.0246
	Feature Reduction	49	45			41		61	58	57	
Dataset 7	Median	ReliefF Relevant	ReliefF Highest 2	Cor Highest	Cor Relevant	Cor Highest 2	ReliefF Highest	RPFS Full	IG Relevant	IG Highest 2	IG Highest
	IQR	0.0065	0.0065	0.0064	0.0065	0.0065	0.0071	0.0124	0.0662	0.0692	0.0899
	Feature Reduction	49		48	48		51		67		77
Dataset 8	Median	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	ReliefF Highest 2	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0134	0.0133	0.0110	0.0143	0.0108	0.0149	0.0150	0.0186	0.0235	0.0210
	Feature Reduction	50		50	53	50			58	63	
Dataset 9	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0075	0.0085	0.0063	0.0068	0.0063	0.0070	0.0070	0.0169	0.0257	0.0147
	Feature Reduction	83		83	83		83		84	85	
Dataset 10	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Highest	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0768	0.0803	0.0778	0.0596	0.0739	0.0713	0.0765	0.0775	0.0665	0.0737
	Feature Reduction	72		84		79	57		77	48	
Dataset 11	Median	IG Relevant	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0106	0.0119	0.0101	0.0132	0.0235	0.0289	0.0349	0.0379	0.0358	0.0159
	Feature Reduction	84	84	84		86			88	89	
Dataset 12	Median	IG Highest 2	IG Highest	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0353	0.0370	0.0373	0.0419	0.0398	0.0439	0.0428	0.0402	0.0479	0.0254
	Feature Reduction		82	76	73		82		86	90	
Dataset 13	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0031	0.0031	0.0032	0.0032	0.0032	0.0020	0.0020	0.0020	0.0042	0.0082
	Feature Reduction	83		83	83		83	83		87	
Dataset 14	Median	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0680	0.0603	0.0643	0.0692	0.0588	0.0825	0.0639	0.0704	0.0850	0.1289
	Feature Reduction	81			84	80	86	83		89	
Dataset 15	Median	Cor Relevant	ReliefF Relevant	Cor Highest 2	IG Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0065	0.0060	0.0070	0.0062	0.0084	0.0063	0.0139	0.0140	0.0119	0.0115
	Feature Reduction	83	83		83			86	86	86	
Dataset 16	Median	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	IG Highest	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0523	0.0579	0.0634	0.0606	0.0658	0.0693	0.0515	0.0487	0.0524	0.0400
	Feature Reduction	81	80			84	87		78	84	
Dataset 17	Median	IG Relevant	IG Highest 2	Cor Relevant	ReliefF Highest 2	ReliefF Relevant	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0077	0.0080	0.0089	0.0083	0.0086	0.0109	0.0179	0.0197	0.0293	0.0143
	Feature Reduction	83		84		83		85	84	86	
Dataset 18	Median	Cor Highest 2	Cor Highest	Cor Relevant	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0775	0.0879	0.0764	0.0703	0.0728	0.0756	0.0575	0.0711	0.0554	0.0624
	Feature Reduction		83	72	77		88		73	38	
Dataset 19	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0122	0.0132	0.0124	0.0124	0.0110	0.0136	0.0157	0.0146	0.0213	0.0140
	Feature Reduction	84		85		85	83		85	86	
Dataset 20	Median	IG Relevant	IG Highest 2	IG Highest	Cor Highest 2	Cor Highest	Cor Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0389	0.0412	0.0468	0.0368	0.0443	0.0282	0.0268	0.0286	0.0294	0.0392
	Feature Reduction	77		84		81	73		70	43	
Dataset 21	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	IG Highest	RPFS Full
	IQR	0.0044	0.0044	0.0044	0.0035	0.0035	0.0035	0.0039	0.0039	0.0040	0.0060
	Feature Reduction	83		84	83	83		83		84	
Dataset 22	Median	Cor Highest 2	Cor Relevant	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0418	0.0481	0.0461	0.0467	0.0484	0.0585	0.0657	0.0643	0.0696	0.1058
	Feature Reduction		81		85	81	86		85	78	
Dataset 23	Median	ReliefF Relevant	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest 2	Cor Highest	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0069	0.0060	0.0060	0.0077	0.0062	0.0100	0.0163	0.0370	0.0132	0.0670
	Feature Reduction	83	83		83	83		88			92
Dataset 24	Median	IG Relevant	Cor Relevant	IG Highest 2	Cor Highest 2	IG Highest	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0456	0.0549	0.0461	0.0612	0.0452	0.0646	0.0538	0.0575	0.0692	0.0465
	Feature Reduction	80	81			84	84		75	82	

Table 5.5: Precision for SVM.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0159	0.0193	0.0196	0.0189	0.0221	0.0244	0.0242	0.0318	0.0256	0.0137
	Feature Reduction	52	52			46		62	63	57	
Dataset 2	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0098	0.0086	0.0103	0.0131	0.0121	0.0164	0.0181	0.0177	0.0215	0.0110
	Feature Reduction	54	54			48	61		63	61	
Dataset 3	Median	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	IG Highest	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0057	0.0057	0.0061	0.0061	0.0071	0.0068	0.0053	0.0064	0.0082	0.0104
	Feature Reduction	50		50		51	51	48		52	
Dataset 4	Median	IG Relevant	ReliefF Relevant	Cor Relevant	IG Highest 2	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0072	0.0062	0.0077	0.0081	0.0093	0.0119	0.0108	0.0173	0.0187	0.0085
	Feature Reduction	50	50	50				57	60	65	
Dataset 5	Median	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0101	0.0103	0.0110	0.0104	0.0119	0.0141	0.0138	0.0158	0.0181	0.0115
	Feature Reduction	48			51	44	49		36	53	
Dataset 6	Median	IG Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0096	0.0092	0.0104	0.0119	0.0103	0.0130	0.0170	0.0178	0.0176	0.0087
	Feature Reduction	45	49			41		61	58	57	
Dataset 7	Median	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full	IG Relevant	IG Highest 2	IG Highest
	IQR	0.0065	0.0064	0.0065	0.0064	0.0064	0.0067	0.0135	0.0298	0.0293	0.0364
	Feature Reduction	48	48		49		51		67		77
Dataset 8	Median	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	IG Highest	ReliefF Highest	Cor Highest 2	Cor Highest	RPFS Full
	IQR	0.0083	0.0085	0.0069	0.0087	0.0073	0.0103	0.0097	0.0099	0.0141	0.0068
	Feature Reduction	50		50		50	53	58		63	
Dataset 9	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0058	0.0058	0.0047	0.0047	0.0047	0.0061	0.0065	0.0111	0.0150	0.0085
	Feature Reduction	83		83	83		83		84	85	
Dataset 10	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	IG Highest 2	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0319	0.0339	0.0393	0.0338	0.0275	0.0266	0.0330	0.0355	0.0292	0.0308
	Feature Reduction	72		84	79		57		77	48	
Dataset 11	Median	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0067	0.0059	0.0056	0.0066	0.0109	0.0167	0.0178	0.0235	0.0225	0.0085
	Feature Reduction	84	84	84		86			88	89	
Dataset 12	Median	IG Highest	IG Highest 2	IG Relevant	Cor Highest 2	Cor Relevant	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0203	0.0191	0.0207	0.0164	0.0171	0.0208	0.0255	0.0248	0.0266	0.0111
	Feature Reduction	82		76		73	82		86	90	
Dataset 13	Median	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0044	0.0049	0.0044	0.0044	0.0044	0.0038	0.0038	0.0038	0.0053	0.0057
	Feature Reduction	83		83	83		83	83		87	
Dataset 14	Median	Cor Highest 2	Cor Relevant	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0304	0.0294	0.0315	0.0351	0.0354	0.0386	0.0404	0.0360	0.0514	0.0394
	Feature Reduction		81	84		80	86		83	89	
Dataset 15	Median	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0049	0.0057	0.0053	0.0048	0.0055	0.0056	0.0068	0.0075	0.0067	0.0067
	Feature Reduction	83		83	83			86	86	86	
Dataset 16	Median	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	ReliefF Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0189	0.0259	0.0232	0.0251	0.0211	0.0313	0.0226	0.0278	0.0257	0.0188
	Feature Reduction	81		80		78	84		87	84	
Dataset 17	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0071	0.0070	0.0054	0.0066	0.0065	0.0073	0.0109	0.0089	0.0146	0.0085
	Feature Reduction	83		84	83			85	84	86	
Dataset 18	Median	Cor Highest	Cor Highest 2	IG Highest 2	IG Relevant	Cor Relevant	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0294	0.0270	0.0301	0.0282	0.0331	0.0368	0.0254	0.0276	0.0256	0.0232
	Feature Reduction	83			77	72	88	73		38	
Dataset 19	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0051	0.0059	0.0061	0.0061	0.0070	0.0062	0.0085	0.0076	0.0121	0.0049
	Feature Reduction	84		85		85	83		85	86	
Dataset 20	Median	IG Highest	IG Relevant	IG Highest 2	Cor Highest 2	Cor Highest	Cor Relevant	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0182	0.0155	0.0186	0.0165	0.0210	0.0135	0.0166	0.0152	0.0168	0.0155
	Feature Reduction	84	77			81	73	70		43	
Dataset 21	Median	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0041	0.0041	0.0041	0.0042	0.0042	0.0043	0.0041	0.0043	0.0042	0.0085
	Feature Reduction	83	83		83		83	84		84	
Dataset 22	Median	Cor Highest 2	Cor Relevant	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0209	0.0190	0.0258	0.0243	0.0230	0.0289	0.0289	0.0271	0.0268	0.0308
	Feature Reduction		81	85		81	86		85	78	
Dataset 23	Median	IG Relevant	IG Highest 2	IG Highest	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0059	0.0059	0.0058	0.0057	0.0062	0.0071	0.0090	0.0147	0.0253	0.0071
	Feature Reduction	83		83	83	83		88		92	
Dataset 24	Median	Cor Relevant	IG Highest	Cor Highest	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0167	0.0215	0.0215	0.0205	0.0186	0.0217	0.0227	0.0214	0.0241	0.0183
	Feature Reduction	81	84	84	80			82		75	

Table 5.6: Recall for SVM.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Highest	IG Highest	RPFS Full
	IQR	0.0213	0.0243	0.0252	0.0254	0.0296	0.0323	0.0383	0.0434	0.0329	0.0210
	Feature Reduction	52	52			46		57	63	62	
Dataset 2	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0184	0.0191	0.0204	0.0241	0.0227	0.0318	0.0365	0.0321	0.0456	0.0178
	Feature Reduction	54	54			48	61		63	61	
Dataset 3	Median	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0047	0.0047	0.0056	0.0050	0.0050	0.0058	0.0071	0.0079	0.0118	0.0126
	Feature Reduction	50		51	50		51	48		52	
Dataset 4	Median	ReliefF Relevant	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest 2	ReliefF Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0082	0.0068	0.0084	0.0067	0.0109	0.0102	0.0169	0.0264	0.0341	0.0159
	Feature Reduction	50	50		50	57			60	65	
Dataset 5	Median	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0158	0.0150	0.0180	0.0156	0.0171	0.0239	0.0204	0.0241	0.0282	0.0187
	Feature Reduction	48			51	44	49		36	53	
Dataset 6	Median	Cor Relevant	IG Relevant	Cor Highest 2	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0196	0.0196	0.0251	0.0196	0.0244	0.0251	0.0396	0.0328	0.0407	0.0170
	Feature Reduction	49	45		41			61	58	57	
Dataset 7	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Relevant	Cor Highest 2	Cor Highest	RPFS Full	IG Relevant	IG Highest 2	IG Highest
	IQR	0.0076	0.0076	0.0085	0.0084	0.0084	0.0077	0.0204	0.0526	0.0522	0.0613
	Feature Reduction	49		51	48		48		67		77
Dataset 8	Median	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	ReliefF Highest 2	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0100	0.0102	0.0095	0.0124	0.0127	0.0106	0.0179	0.0129	0.0252	0.0127
	Feature Reduction	50		50	53	50			58	63	
Dataset 9	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0056	0.0056	0.0040	0.0040	0.0040	0.0056	0.0062	0.0143	0.0195	0.0126
	Feature Reduction	83		83	83		83		84	85	
Dataset 10	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	IG Highest 2	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0489	0.0466	0.0573	0.0459	0.0434	0.0416	0.0439	0.0521	0.0423	0.0449
	Feature Reduction	72		84	79		57		77	48	
Dataset 11	Median	IG Relevant	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0087	0.0074	0.0088	0.0090	0.0205	0.0258	0.0349	0.0389	0.0399	0.0101
	Feature Reduction	84	84	84		86			88	89	
Dataset 12	Median	IG Highest	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0382	0.0347	0.0404	0.0313	0.0331	0.0347	0.0490	0.0503	0.0473	0.0174
	Feature Reduction	82		76	73		82	86		90	
Dataset 13	Median	ReliefF Relevant	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0019	0.0016	0.0016	0.0016	0.0020	0.0020	0.0020	0.0020	0.0039	0.0065
	Feature Reduction	83	83	83		83	83			87	
Dataset 14	Median	Cor Highest 2	Cor Highest	Cor Relevant	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0402	0.0456	0.0394	0.0496	0.0477	0.0469	0.0613	0.0478	0.0712	0.0550
	Feature Reduction		84	81		80	86		83	89	
Dataset 15	Median	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0035	0.0038	0.0038	0.0043	0.0037	0.0046	0.0080	0.0068	0.0075	0.0103
	Feature Reduction	83	83	83				86	86	86	
Dataset 16	Median	Cor Relevant	Cor Highest 2	IG Highest 2	IG Relevant	IG Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0360	0.0469	0.0550	0.0499	0.0554	0.0439	0.0496	0.0515	0.0582	0.0320
	Feature Reduction	81			80	84	78		84	87	
Dataset 17	Median	Cor Relevant	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0069	0.0077	0.0078	0.0106	0.0105	0.0092	0.0146	0.0138	0.0201	0.0128
	Feature Reduction	84	83		83			85	84	86	
Dataset 18	Median	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	Cor Relevant	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0479	0.0558	0.0596	0.0546	0.0602	0.0666	0.0517	0.0465	0.0469	0.0380
	Feature Reduction		83		77	72	88	73		38	
Dataset 19	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0082	0.0086	0.0095	0.0095	0.0113	0.0106	0.0153	0.0112	0.0188	0.0124
	Feature Reduction	84		85		85	83	85		86	
Dataset 20	Median	IG Highest	IG Relevant	IG Highest 2	Cor Highest 2	Cor Relevant	Cor Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0394	0.0300	0.0396	0.0343	0.0262	0.0419	0.0360	0.0311	0.0329	0.0315
	Feature Reduction	84	77			73	81	70		43	
Dataset 21	Median	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0019	0.0019	0.0019	0.0019	0.0019	0.0023	0.0023	0.0023	0.0020	0.0109
	Feature Reduction	83	83		83		83		84	84	
Dataset 22	Median	Cor Highest 2	Cor Relevant	Cor Highest	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0397	0.0400	0.0440	0.0410	0.0405	0.0438	0.0489	0.0496	0.0430	0.0607
	Feature Reduction		81	85	81		86	85		78	
Dataset 23	Median	ReliefF Relevant	Cor Relevant	IG Relevant	IG Highest 2	IG Highest	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0038	0.0031	0.0038	0.0038	0.0040	0.0055	0.0103	0.0250	0.0452	0.0102
	Feature Reduction	83	83	83		83		88		92	
Dataset 24	Median	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest	Cor Highest 2	ReliefF Highest	ReliefF Relevant	ReliefF Highest 2	RPFS Full
	IQR	0.0352	0.0381	0.0301	0.0371	0.0384	0.0331	0.0454	0.0495	0.0447	0.0439
	Feature Reduction	80		81	84	84		82	75		

The first notable observation for all four evaluation measures is the fact that the reduced models consistently perform better than the full model for all datasets except for Dataset 7. Dataset 7 is characterised by the following: $k = 10$, $p - k = 10$, $q = 6$, $\rho = 0$, signal strength = 100, the entries of the vector of density are not constant, number of training instances = 80, and number

of test instances = 10 000. Based on these characteristics, it is unclear why the results based on the analysis of Dataset 7 differs from the other datasets.

The second observation is that the procedures using the correlation coefficient and IG as relevance measures seem to consistently outrank those based on ReliefF. Finally, it is interesting to note that, in general, a procedure that performs well in terms of Precision typically performs relatively poorly in terms of Recall. This is not the case for the SVM classifier.

To summarise the tables, the Method of Pairwise Comparisons is applied again, but now using the median of the IQR over the 24 datasets as a measure of variation. Darker shades of green correspond to larger IQRs. The resulting Table 5.7 provides a ranking of the procedures over all 24 datasets.

Table 5.7: Method of Pairwise Comparisons for all 24 datasets – SVM.

Rank	Hamming Loss		One-Error		Precision		Recall	
	Technique	Median IQR	Technique	Median IQR	Technique	Median IQR	Technique	Median IQR
1	Cor Relevant	0.00738	Cor Relevant	0.01859	Cor Relevant	0.00845	Cor Relevant	0.01426
2	IG Relevant	0.00937	IG Relevant	0.02124	IG Relevant	0.00908	IG Relevant	0.01810
3	Cor Highest 2	0.00954	Cor Highest 2	0.02056	Cor Highest 2	0.01013	Cor Highest 2	0.01647
4	IG Highest 2	0.01198	IG Highest 2	0.02925	IG Highest 2	0.01252	IG Highest 2	0.02423
5	ReliefF Relevant	0.00795	Cor Highest	0.03240	Cor Highest	0.01547	ReliefF Relevant	0.01509
6	Cor Highest	0.01453	ReliefF Relevant	0.01799	ReliefF Relevant	0.00870	Cor Highest	0.02578
7	IG Highest	0.01565	IG Highest	0.03649	ReliefF Highest 2	0.01425	IG Highest	0.03243
8	ReliefF Highest 2	0.01474	ReliefF Highest 2	0.02845	IG Highest	0.01778	ReliefF Highest 2	0.02506
9	ReliefF Highest	0.01890	ReliefF Highest	0.03587	ReliefF Highest	0.01836	ReliefF Highest	0.03711
10	RPFS Full	0.00916	RPFS Full	0.02191	RPFS Full	0.01070	RPFS Full	0.01719

From Table 5.7 it can be seen that four procedures consistently perform better than the rest, namely using only the relevant features identified by the correlation coefficient and IG, and using the highest two ranked features from every feature group – once again identified by the correlation coefficient and IG as relevance measures. The procedures based on only the relevant features also have low variation. The performances of the procedures are fairly consistent across the four evaluation measures, and all FS procedures rank higher than the full model.

In order to assess the procedures based on the properties of the datasets that are of interest, namely the signal strength, the number of irrelevant features, the number of training instances,

the label dependence, and the vector of label densities, the 24 datasets are grouped according to these properties and the results are presented in Tables 5.8 to 5.13.

In Table 5.8, the effect of the signal strength on the performances of the procedures is investigated. Irrespective of the signal level, the procedure that uses all the relevant features identified by the correlation coefficient as relevance measure ranks highest with a low variation for all four evaluation measures. Other procedures that consistently rank high are the procedures Correlation Highest 2, IG, Relevant and IG Highest 2. The procedure ReliefF Highest consistently ranks as the poorest of the nine FS approaches. The ranking of the procedure ReliefF Relevant improves for all four evaluation measures when the signal strengthens.

In order to investigate the influence of the number of noise features on the performances of the FS procedures, the datasets are grouped according to the number of irrelevant features and the results are displayed in Table 5.9. Once again, Correlation Relevant ranks highest and ReliefF Highest is ranked lowest irrespective of the number of irrelevant features. It is interesting that the ranking of Correlation Highest improves as the level of noise increases.

Tables 5.10 and 5.11 represent the rankings of the procedures according to the number of training instances. It is interesting that the procedure Correlation Highest 2 ranks highest for the scenario where the number of training instances is small, namely $N = 30$, for three of the evaluation measures (Hamming-loss, Precision, and Recall), and ranks second highest for One-error. The procedures based on the Correlation Relevant rank highest for Hamming-loss, One-error, Precision, and Recall when $N = 80$.

When the procedures based on the label dependence are compared in Table 5.12, the performance of IG Relevant improves when there is some degree of label dependence. Surprisingly, the methods based on the correlation coefficient perform better when there is no correlation present amongst the labels. Once again, ReliefF Highest consistently ranks last.

Irrespective of the vector of densities used, the procedures that use the correlation coefficient and IG as relevance measure on all relevant features, rank highest with a low variation for all four evaluation measures. The procedures based on ReliefF Highest consistently ranks as the poorest of the nine FS procedures. See Table 5.13.

In general, the FS procedures based on the correlation coefficient and IG, including the relevant features and the two highest ranked features from each of the feature groups, are ranked highest

with fairly low variation. The FS methods based on ReliefF show higher variation and consistently rank lower than the other procedures.

Table 5.8: Method of Pairwise Comparisons for the signal level – SVM.

	Hamming Loss				One-Error				Precision				Recall			
Rank	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR
1	Cor Relevant	0.00915	Cor Relevant	0.00531	Cor Relevant	0.02116	Cor Relevant	0.00921	Cor Relevant	0.00999	Cor Relevant	0.00688	Cor Relevant	0.01899	Cor Relevant	0.00750
2	IG Relevant	0.01029	IG Relevant	0.00561	IG Relevant	0.02219	IG Relevant	0.01388	IG Relevant	0.01075	Cor Highest 2	0.00822	IG Relevant	0.01932	IG Relevant	0.00842
3	Cor Highest 2	0.01059	Cor Highest 2	0.00641	Cor Highest 2	0.02650	ReliefF Relevant	0.00895	Cor Highest 2	0.01036	IG Relevant	0.00775	Cor Highest 2	0.02276	IG Highest 2	0.00930
4	IG Highest 2	0.01364	IG Highest 2	0.00582	IG Highest 2	0.02996	Cor Highest 2	0.01249	IG Highest 2	0.01488	IG Highest 2	0.00832	IG Highest 2	0.02491	ReliefF Relevant	0.00785
5	Cor Highest	0.01643	ReliefF Relevant	0.00558	Cor Highest	0.03970	IG Highest 2	0.01516	Cor Highest	0.01668	ReliefF Relevant	0.00630	Cor Highest	0.03326	Cor Highest 2	0.00926
6	ReliefF Relevant	0.01315	IG Highest	0.00775	ReliefF Highest 2	0.02993	IG Highest	0.01926	ReliefF Relevant	0.01394	Cor Highest	0.01153	ReliefF Relevant	0.02342	IG Highest	0.01168
7	ReliefF Highest 2	0.01531	Cor Highest	0.01019	ReliefF Relevant	0.02640	Cor Highest	0.01986	IG Highest	0.01802	IG Highest	0.01059	IG Highest	0.03286	Cor Highest	0.01776
8	IG Highest	0.01714	ReliefF Highest 2	0.00849	IG Highest	0.03745	ReliefF Highest 2	0.02201	ReliefF Highest 2	0.01647	ReliefF Highest 2	0.01030	ReliefF Highest 2	0.03173	ReliefF Highest 2	0.01375
9	ReliefF Highest	0.01890	ReliefF Highest	0.01561	ReliefF Highest	0.03587	ReliefF Highest	0.02813	ReliefF Highest	0.01979	ReliefF Highest	0.01417	ReliefF Highest	0.03911	ReliefF Highest	0.02352
10	RPFS Full	0.00921	RPFS Full	0.00876	RPFS Full	0.02405	RPFS Full	0.01801	RPFS Full	0.01108	RPFS Full	0.00945	RPFS Full	0.01756	RPFS Full	0.01428

Table 5.9: Method of Pairwise Comparisons for the number of irrelevant features – SVM.

	Hamming Loss				One-Error				Precision				Recall			
Rank	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR
1	Cor Relevant	0.00738	Cor Relevant	0.00862	Cor Relevant	0.01859	Cor Relevant	0.02032	Cor Relevant	0.00845	Cor Relevant	0.00982	Cor Relevant	0.01426	Cor Relevant	0.01782
2	IG Relevant	0.00937	IG Relevant	0.01026	IG Relevant	0.02124	IG Relevant	0.02474	IG Relevant	0.00908	Cor Highest 2	0.01182	IG Relevant	0.01810	IG Relevant	0.01937
3	Cor Highest 2	0.00954	Cor Highest 2	0.01046	Cor Highest 2	0.02056	Cor Highest 2	0.02500	Cor Highest 2	0.01013	IG Relevant	0.01111	Cor Highest 2	0.01647	Cor Highest 2	0.02127
4	IG Highest 2	0.01038	IG Highest 2	0.01575	IG Highest 2	0.02553	IG Highest 2	0.03210	IG Highest 2	0.01149	Cor Highest	0.01767	IG Highest 2	0.02102	IG Highest 2	0.03025
5	ReliefF Relevant	0.00795	Cor Highest	0.01474	ReliefF Relevant	0.01779	Cor Highest	0.03660	ReliefF Relevant	0.00858	IG Highest 2	0.01762	ReliefF Relevant	0.01455	Cor Highest	0.02764
6	ReliefF Highest 2	0.01174	IG Highest	0.01759	ReliefF Highest 2	0.02670	IG Highest	0.03745	ReliefF Highest 2	0.01248	IG Highest	0.01924	ReliefF Highest 2	0.01866	IG Highest	0.03766
7	Cor Highest	0.01453	ReliefF Relevant	0.01082	Cor Highest	0.03090	ReliefF Relevant	0.02151	Cor Highest	0.01521	ReliefF Relevant	0.01195	Cor Highest	0.02578	ReliefF Relevant	0.02178
8	IG Highest	0.01445	ReliefF Highest 2	0.01593	IG Highest	0.03141	ReliefF Highest 2	0.03596	IG Highest	0.01590	ReliefF Highest 2	0.01647	IG Highest	0.02799	ReliefF Highest 2	0.03302
9	ReliefF Highest	0.01739	ReliefF Highest	0.02137	ReliefF Highest	0.03400	ReliefF Highest	0.04184	ReliefF Highest	0.01784	ReliefF Highest	0.02260	ReliefF Highest	0.03112	ReliefF Highest	0.04259
10	RPFS Full	0.00913	RPFS Full	0.00921	RPFS Full	0.02191	RPFS Full	0.02065	RPFS Full	0.01070	RPFS Full	0.00982	RPFS Full	0.01738	RPFS Full	0.01511

Table 5.10: Method of Pairwise Comparisons for the number of training instances: Hamming-loss and One-error – SVM.

	Hamming Loss						One-Error					
Rank	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR
1	Cor Highest 2	0.02339	Cor Relevant	0.00738	IG Relevant	0.00303	Cor Relevant	0.05364	Cor Relevant	0.01859	IG Relevant	0.00683
2	Cor Relevant	0.01959	IG Relevant	0.00937	Cor Relevant	0.00317	Cor Highest 2	0.06043	IG Relevant	0.02124	ReliefF Relevant	0.00696
3	IG Highest 2	0.02349	Cor Highest 2	0.00954	ReliefF Relevant	0.00350	IG Highest 2	0.05281	Cor Highest 2	0.02056	Cor Relevant	0.00706
4	IG Relevant	0.02221	IG Highest 2	0.01038	Cor Highest 2	0.00358	IG Relevant	0.05310	IG Highest 2	0.02553	IG Highest 2	0.00718
5	Cor Highest	0.02741	ReliefF Relevant	0.00795	IG Highest 2	0.00342	Cor Highest	0.06688	ReliefF Relevant	0.01779	Cor Highest 2	0.00847
6	IG Highest	0.02946	ReliefF Highest 2	0.01174	ReliefF Highest 2	0.00532	IG Highest	0.06211	ReliefF Highest 2	0.02670	Cor Highest	0.01243
7	ReliefF Highest 2	0.02640	Cor Highest	0.01453	Cor Highest	0.00504	ReliefF Highest 2	0.05566	Cor Highest	0.03090	ReliefF Highest 2	0.00835
8	ReliefF Highest	0.02689	IG Highest	0.01445	IG Highest	0.00508	ReliefF Relevant	0.05644	IG Highest	0.03141	IG Highest	0.01324
9	ReliefF Relevant	0.02518	ReliefF Highest	0.01739	ReliefF Highest	0.00930	ReliefF Highest	0.06674	ReliefF Highest	0.03400	ReliefF Highest	0.01830
10	RPFS Full	0.02008	RPFS Full	0.00913	RPFS Full	0.00543	RPFS Full	0.05446	RPFS Full	0.02191	RPFS Full	0.01355

Table 5.11: Method of Pairwise Comparisons for the number of training instances: Precision and Recall – SVM.

	Precision						Recall					
Rank	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR
1	Cor Highest 2	0.02339	Cor Relevant	0.00845	ReliefF Relevant	0.00575	Cor Highest 2	0.03994	Cor Relevant	0.01426	IG Relevant	0.00469
2	Cor Relevant	0.01895	IG Relevant	0.00908	Cor Relevant	0.00515	Cor Relevant	0.03767	IG Relevant	0.01810	Cor Relevant	0.00389
3	Cor Highest	0.02679	Cor Highest 2	0.01013	IG Relevant	0.00552	IG Highest 2	0.04196	Cor Highest 2	0.01647	ReliefF Relevant	0.00472
4	IG Highest	0.03008	IG Highest 2	0.01149	Cor Highest 2	0.00592	IG Relevant	0.04130	IG Highest 2	0.02102	Cor Highest 2	0.00490
5	IG Highest 2	0.02473	ReliefF Relevant	0.00858	ReliefF Highest 2	0.00640	Cor Highest	0.04482	ReliefF Relevant	0.01455	IG Highest 2	0.00500
6	IG Relevant	0.02313	ReliefF Highest 2	0.01248	IG Highest 2	0.00594	IG Highest	0.04485	ReliefF Highest 2	0.01866	Cor Highest	0.00917
7	ReliefF Highest 2	0.02656	Cor Highest	0.01521	Cor Highest	0.00690	ReliefF Highest	0.05017	Cor Highest	0.02578	ReliefF Highest 2	0.00804
8	ReliefF Highest	0.02617	IG Highest	0.01590	IG Highest	0.00715	ReliefF Highest 2	0.04805	IG Highest	0.02799	IG Highest	0.01072
9	ReliefF Relevant	0.02520	ReliefF Highest	0.01784	ReliefF Highest	0.00997	ReliefF Relevant	0.04541	ReliefF Highest	0.03112	ReliefF Highest	0.01405
10	RPFS Full	0.02100	RPFS Full	0.01070	RPFS Full	0.00778	RPFS Full	0.04094	RPFS Full	0.01738	RPFS Full	0.01059

Table 5.12: Method of Pairwise Comparisons for the label dependence – SVM.

	Hamming Loss				One-Error				Precision				Recall			
Rank	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR
1	Cor Relevant	0.00765	IG Relevant	0.00731	Cor Relevant	0.01583	IG Relevant	0.01785	Cor Relevant	0.00832	IG Relevant	0.00843	Cor Relevant	0.01207	IG Relevant	0.01456
2	Cor Highest 2	0.00765	IG Highest 2	0.01198	Cor Highest 2	0.01643	Cor Relevant	0.01859	Cor Highest 2	0.00879	Cor Relevant	0.00845	Cor Highest 2	0.01213	Cor Relevant	0.01558
3	Cor Highest	0.01105	Cor Relevant	0.00738	Cor Highest	0.02780	IG Highest 2	0.02925	Cor Highest	0.01245	Cor Highest 2	0.01015	Cor Highest	0.01780	IG Highest 2	0.02423
4	IG Highest 2	0.01461	Cor Highest 2	0.00957	IG Highest 2	0.02589	Cor Highest 2	0.02056	IG Relevant	0.01560	IG Highest 2	0.01252	IG Relevant	0.02071	Cor Highest 2	0.01916
5	IG Relevant	0.01457	ReliefF Relevant	0.00750	IG Relevant	0.02463	ReliefF Relevant	0.01799	IG Highest 2	0.01497	ReliefF Relevant	0.00858	IG Highest 2	0.02170	ReliefF Relevant	0.01509
6	IG Highest	0.01899	IG Highest	0.01565	IG Highest	0.03640	IG Highest	0.03649	IG Highest	0.01959	Cor Highest	0.01668	IG Highest	0.02842	IG Highest	0.03243
7	ReliefF Relevant	0.01088	Cor Highest	0.01600	ReliefF Relevant	0.01918	Cor Highest	0.03626	ReliefF Relevant	0.01147	IG Highest	0.01778	ReliefF Relevant	0.01735	Cor Highest	0.02910
8	ReliefF Highest 2	0.01053	ReliefF Highest 2	0.01531	ReliefF Highest 2	0.01698	ReliefF Highest 2	0.03065	ReliefF Highest 2	0.01040	ReliefF Highest 2	0.01495	ReliefF Highest 2	0.01545	ReliefF Highest 2	0.02810
9	ReliefF Highest	0.01417	ReliefF Highest	0.02063	ReliefF Highest	0.02784	ReliefF Highest	0.03675	ReliefF Highest	0.01456	ReliefF Highest	0.02009	ReliefF Highest	0.02123	ReliefF Highest	0.04032
10	RPFS Full	0.01193	RPFS Full	0.00795	RPFS Full	0.01711	RPFS Full	0.02316	RPFS Full	0.01252	RPFS Full	0.00861	RPFS Full	0.01952	RPFS Full	0.01645

Table 5.13: Method of Pairwise Comparisons for different vectors of density – SVM.

	Hamming Loss				One-Error				Precision				Recall			
Rank	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR
1	Cor Relevant	0.00666	IG Relevant	0.00949	Cor Relevant	0.01859	IG Relevant	0.02124	Cor Relevant	0.00878	Cor Relevant	0.00822	Cor Relevant	0.01288	Cor Relevant	0.01426
2	IG Relevant	0.00767	Cor Relevant	0.00754	IG Relevant	0.01869	Cor Relevant	0.01598	Cor Highest 2	0.00983	Cor Highest 2	0.01013	IG Relevant	0.01389	IG Relevant	0.01834
3	Cor Highest 2	0.00830	IG Highest 2	0.01038	Cor Highest 2	0.02056	Cor Highest 2	0.01848	IG Relevant	0.00786	IG Relevant	0.01075	Cor Highest 2	0.01530	Cor Highest 2	0.01647
4	IG Highest 2	0.01364	Cor Highest 2	0.00964	IG Highest 2	0.02925	IG Highest 2	0.02614	IG Highest 2	0.01488	Cor Highest	0.01430	IG Highest 2	0.02473	IG Highest 2	0.02120
5	ReliefF Relevant	0.00835	Cor Highest	0.01326	Cor Highest	0.03626	Cor Highest	0.02780	ReliefF Relevant	0.00912	IG Highest 2	0.01149	ReliefF Relevant	0.01577	Cor Highest	0.02261
6	Cor Highest	0.01600	IG Highest	0.01445	ReliefF Relevant	0.01779	ReliefF Relevant	0.01849	ReliefF Highest 2	0.01795	IG Highest	0.01597	Cor Highest	0.02910	ReliefF Relevant	0.01509
7	ReliefF Highest 2	0.01634	ReliefF Relevant	0.00795	ReliefF Highest 2	0.03315	IG Highest	0.03141	Cor Highest	0.01681	ReliefF Relevant	0.00870	IG Highest	0.03250	IG Highest	0.02834
8	IG Highest	0.01828	ReliefF Highest 2	0.01378	IG Highest	0.03745	ReliefF Highest 2	0.02625	IG Highest	0.01900	ReliefF Highest 2	0.01340	ReliefF Highest 2	0.03363	ReliefF Highest 2	0.02272
9	ReliefF Highest	0.02217	ReliefF Highest	0.01655	ReliefF Highest	0.03956	ReliefF Highest	0.03033	ReliefF Highest	0.02201	ReliefF Highest	0.01710	ReliefF Highest	0.03911	ReliefF Highest	0.03206
10	RPFS Full	0.00916	RPFS Full	0.00948	RPFS Full	0.02101	RPFS Full	0.02280	RPFS Full	0.01070	RPFS Full	0.01011	RPFS Full	0.01663	RPFS Full	0.01785

Feature Reduction

A comparison of the performances of the FS procedures will not be complete without a comparison in terms of *Feature Reduction* (see Section 4.4.2). The Feature Reduction property provides an indication of how many features are excluded by an FS technique, *i.e.* the higher the value of the Feature Reduction measure, the fewer features were included. Results that were obtained in the empirical study in this regard are presented in Figures 5.6 to 5.8. This is done for all 24 datasets separately, for all three relevance measures, and for the models Relevant and Highest Ranked from each feature group.

For Datasets 1 to 8, where there are ten relevant and only ten irrelevant features included in each of the datasets, one would expect lower values of Feature Reduction for Datasets 1 – 8 than for Datasets 9 – 24. The results presented in Figure 5.6 confirm this expectation when compared to Figures 5.7 and 5.8. Most values of the Feature Reduction for Dataset 1 – 8 lie between 40 and 60, with one exception, namely for IG Highest used in Dataset 7 where the Feature Reduction is approximately 80. This most probably explains the poor performance of IG Highest in Tables 5.3 to 5.5.

For the remaining datasets, Datasets 9 to 24, ten relevant and 50 irrelevant features are included in each. One would expect the FS techniques to produce higher values of Feature Reduction due to the higher proportion of noise present in these datasets. This is confirmed in Figures 5.7 and 5.8. It should be noted that the values of Feature Reduction appearing in these figures are fairly close for most of the datasets, with the exception of Datasets 10, 18, and 20. For these three datasets, ReliefF Relevant includes more features compared to its competitors. A closer look at the characteristics of these three datasets is required.

There are four datasets where a weak signal (a signal of ten) is combined with a small number of training instances, $N = 30$, namely Dataset 10, 12, 18, and 20. The results for Dataset 10, 18 and 20 shown in Figures 5.7 and 5.8, seems to provide some evidence that the procedure ReliefF Relevant includes more features when this combination of characteristics is present in a dataset.

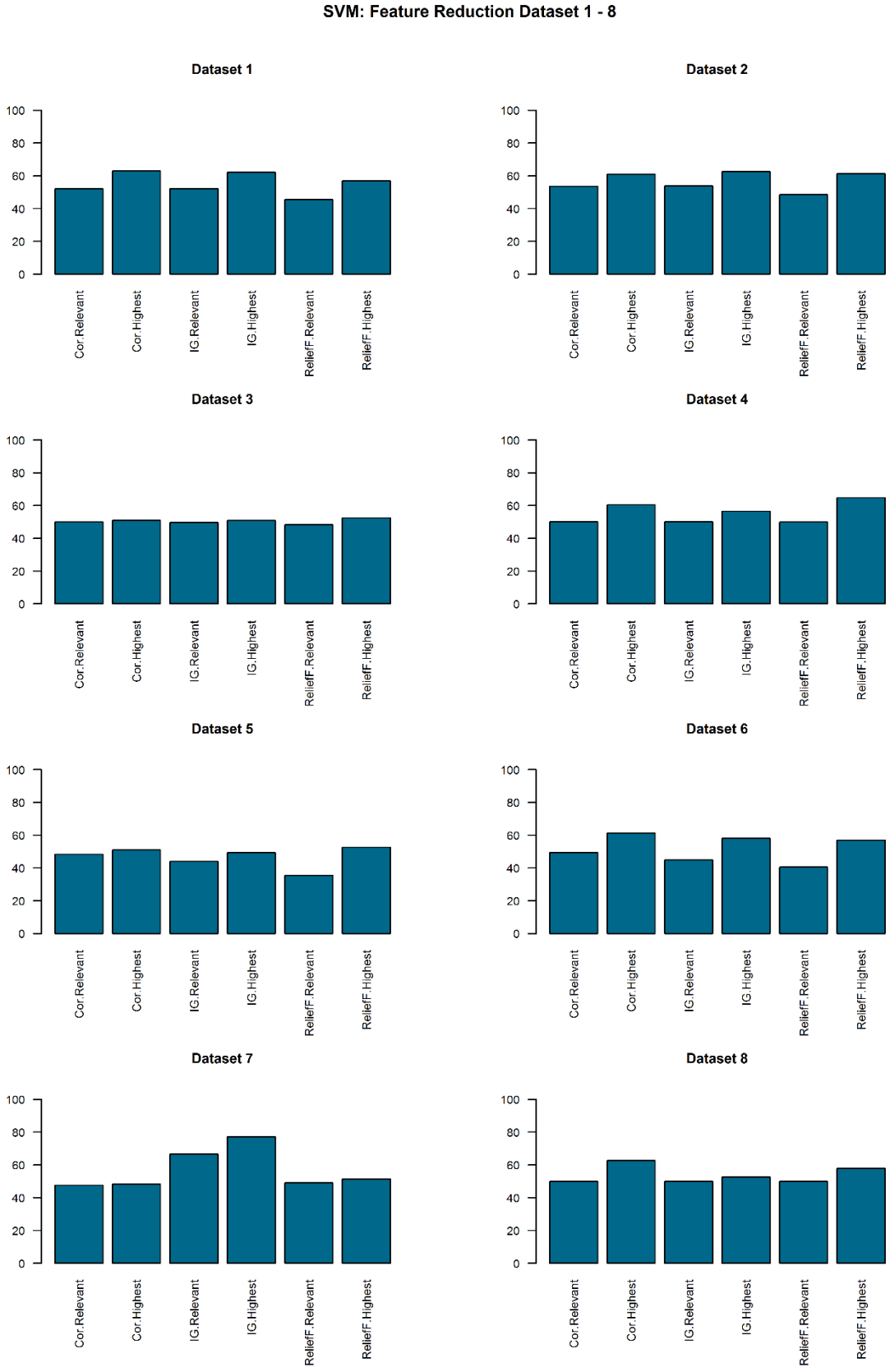


Figure 5.6 Comparison of feature reduction achieved for the SVM classifier: Dataset 1 – 8.

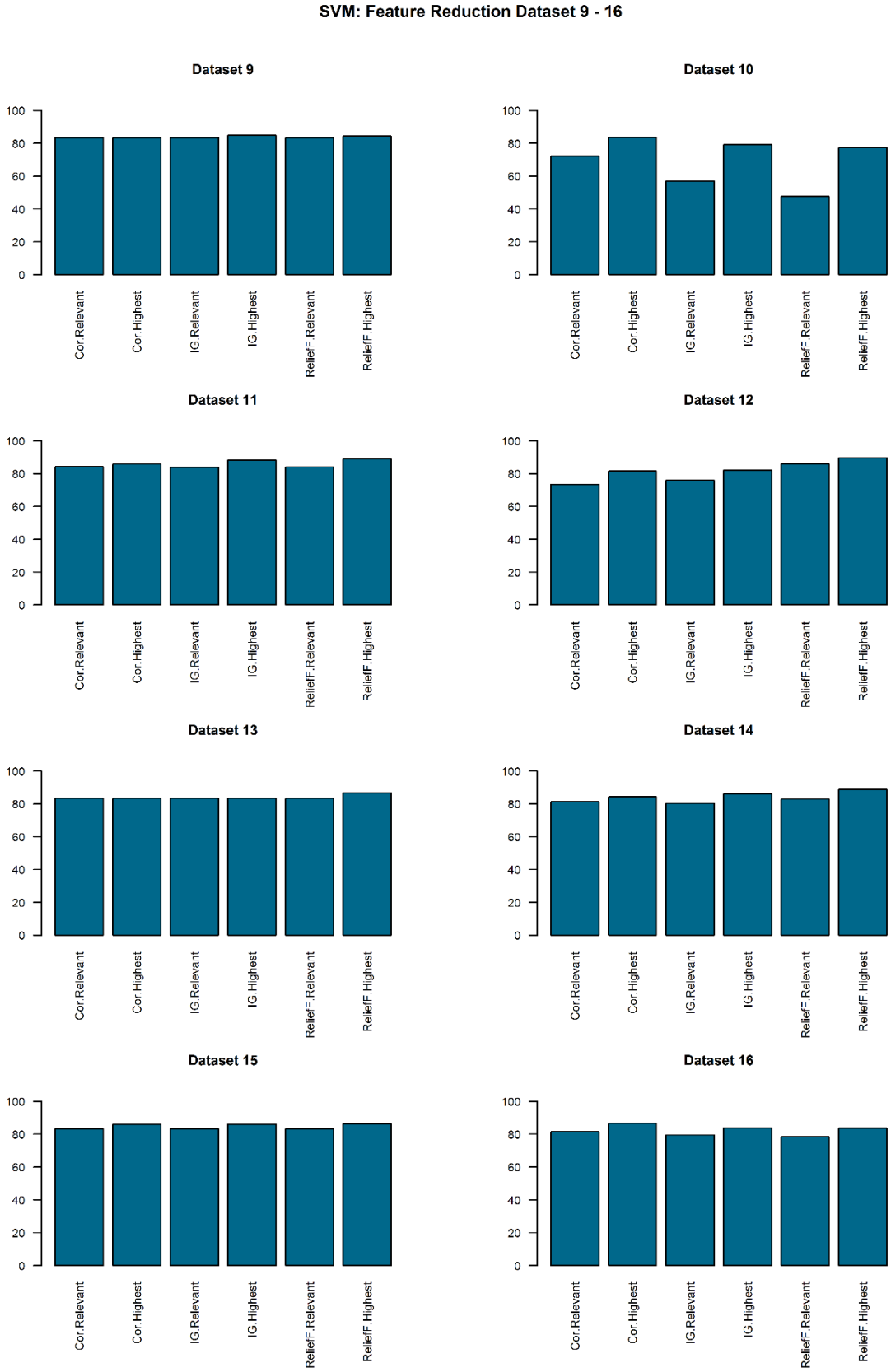


Figure 5.7 Comparison of feature reduction achieved for the SVM classifier: Dataset 9 – 16.

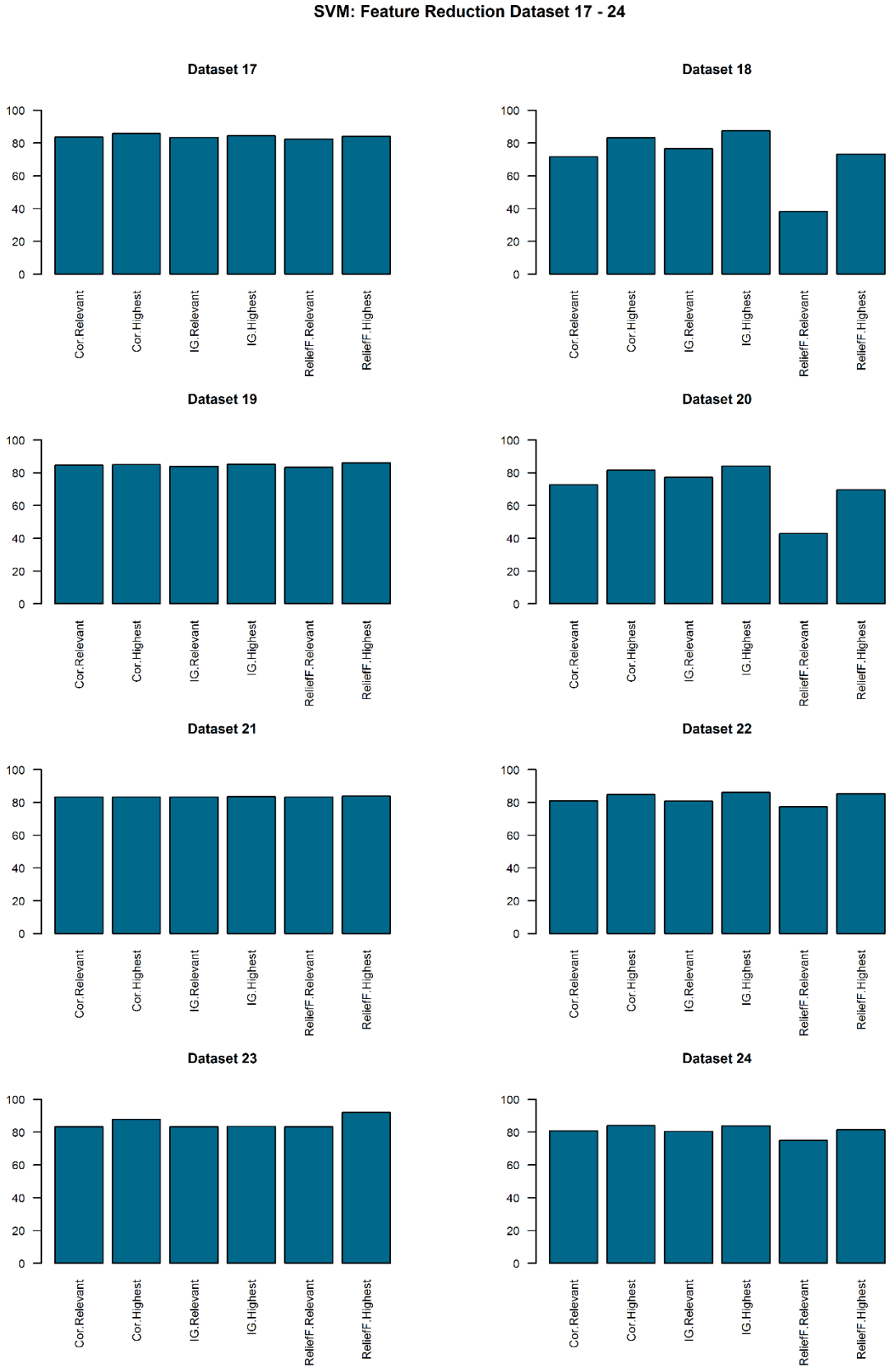


Figure 5.8 Comparison of feature reduction achieved for the SVM classifier: Dataset 17 – 24.

Some important observations regarding the results based on the SVM classifier are:

1) The correlation coefficient as relevance measure does not receive as much attention as IG or ReliefF in research pertaining to multi-label FS. Based on the results presented in this section, it generally performs better than both IG and ReliefF when only the relevant features are included.
2) The inclusion of only the highest ranked feature from each feature group leads to vastly reduced feature sets. However, the results of these reduced feature sets are better than the results obtained using the full set of features.
3) The SVM classifier is sensitive to smaller values of density, typically values smaller than 0.2 for datasets where the number of training instances are also relatively small, *i.e.* $N = 30$. Cognisance should be taken of this fact, especially when considering the problem of unbalanced (binary) classification.

In the next section, the results for the scenarios where the XGBoost classifier was used, will be presented and discussed.

5.4.2 Relevance pattern feature selection – Extreme gradient boosting classifier

In this section, the performance of the RPFS procedures implementing the XGBoost classifier is investigated. In Figures 5.9 to 5.12 the results of the 100 repetitions are summarised using boxplots for each of the FS procedures. As in Section 5.4.1 above, Dataset 1 is used as an example. Similar graphs have been generated for all 24 synthetic datasets. Due to space constraints, the other graphs are available in Appendix C[6].

[6] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

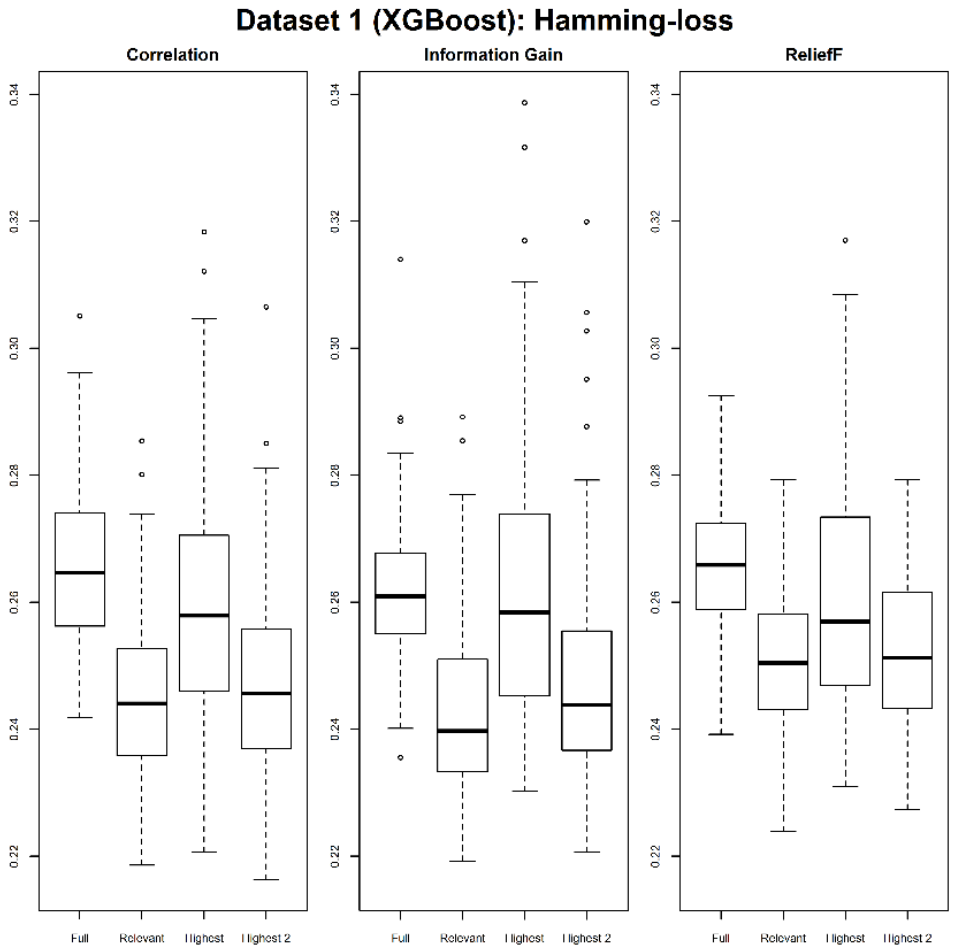


Figure 5.9 Comparison of Hamming-loss using the XGBoost classifier: Dataset 1.

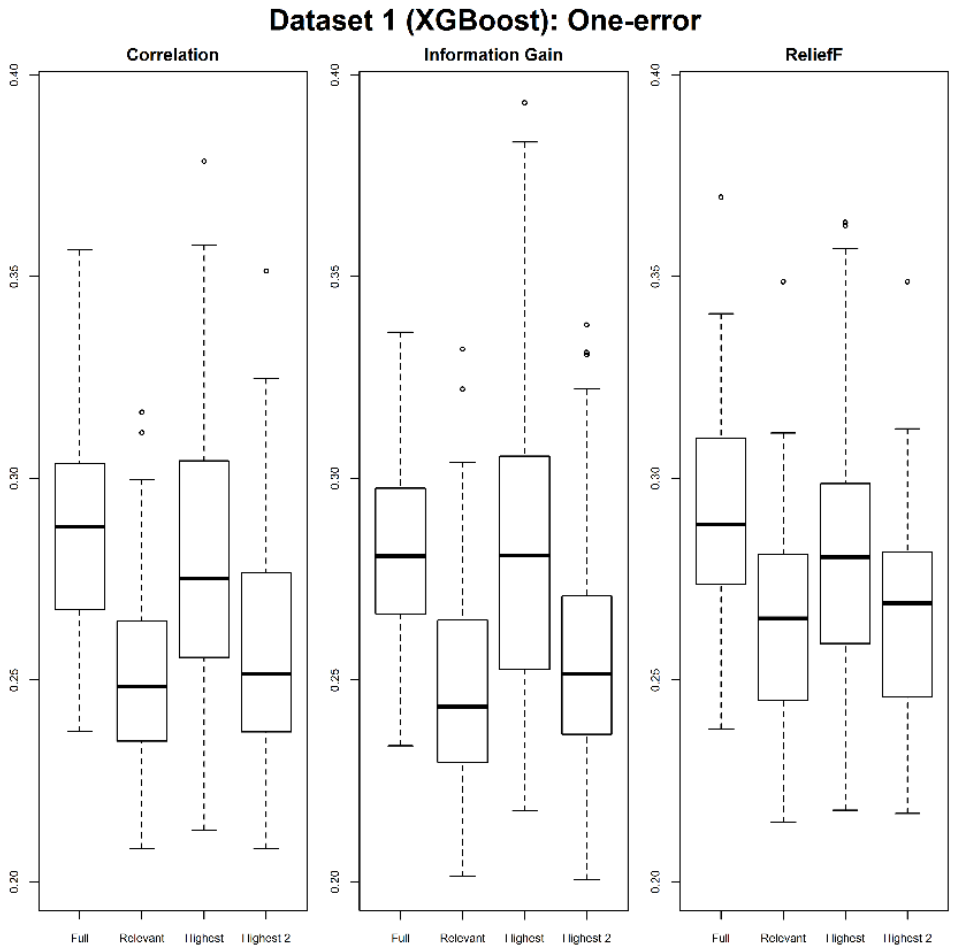


Figure 5.10 Comparison of One-error using the XGBoost classifier: Dataset 1.

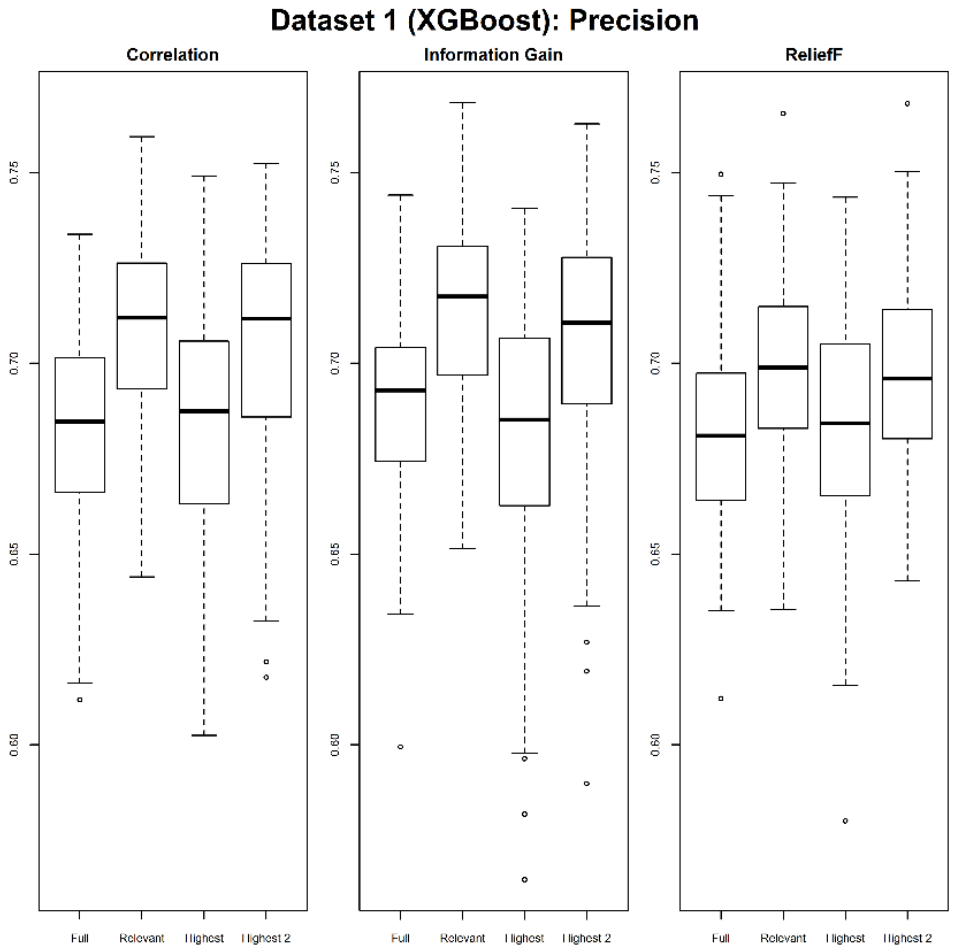


Figure 5.11 Comparison of Precision using the XGBoost classifier: Dataset 1.

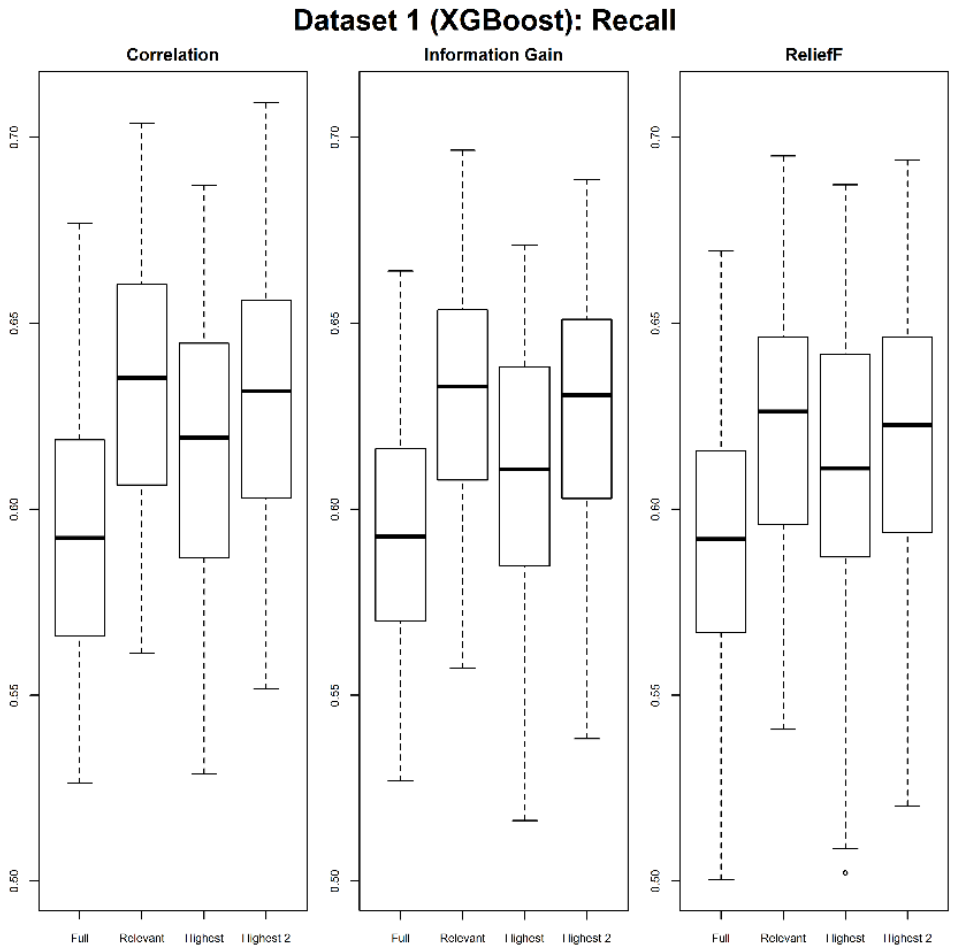


Figure 5.12 Comparison of Recall using the XGBoost classifier: Dataset 1.

With the exception of Precision, the procedures based on the reduced models outperform the full model, with the procedure that includes all relevant features having the best performance. The medians of the reduced model Highest 2 are similar to those of the procedures that include all relevant features. There seems to be more variation in the model that only uses the highest ranked feature from every feature group.

To compare the results across all four evaluation measures, Figure 5.13 provides a summary by combining Figures 5.9 to 5.12 in a single graph.

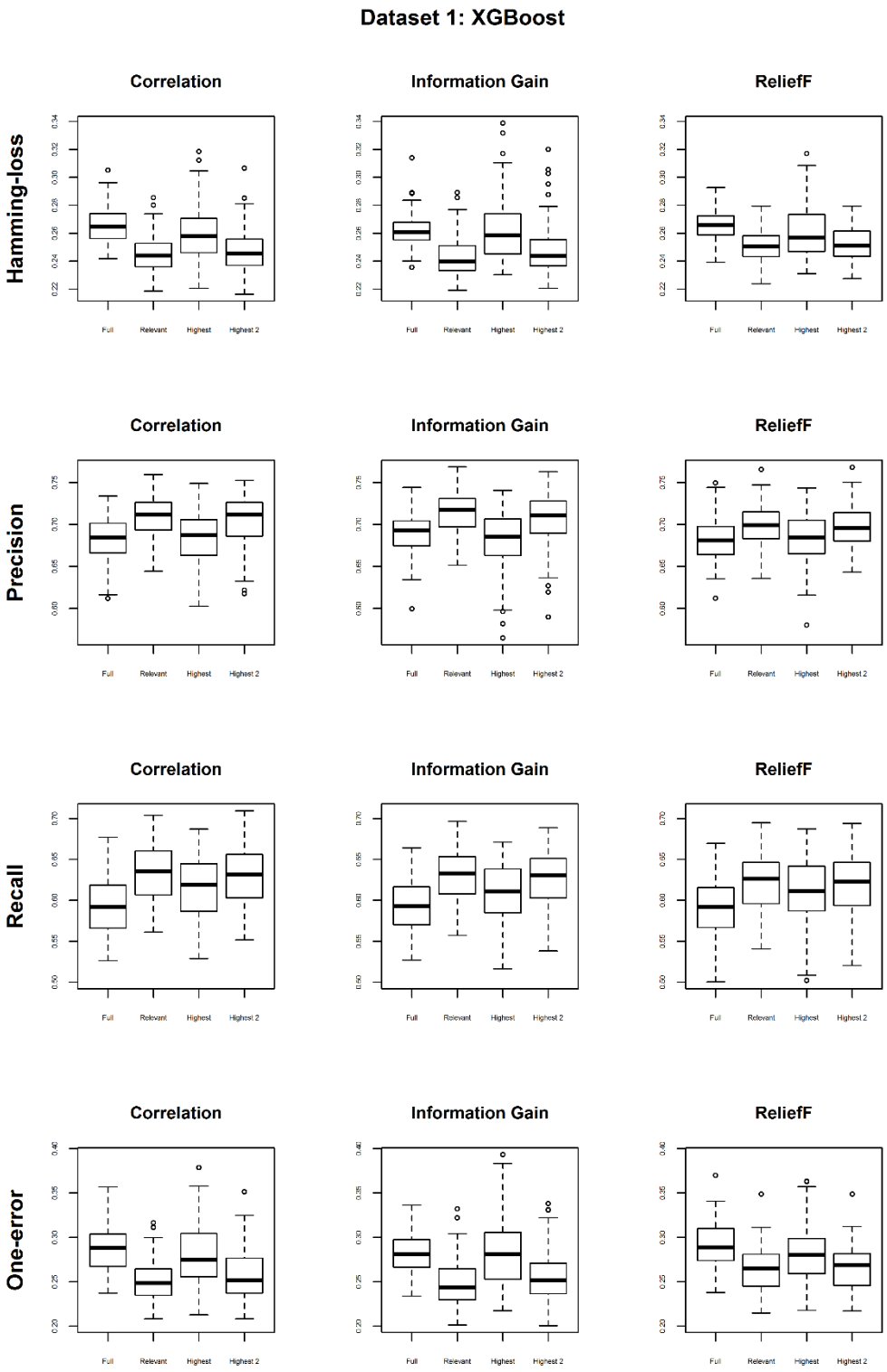


Figure 5.13 Summary of results for the XGBoost classifier: Dataset 1.

As with the interpretation of Figure 5.5 for the SVM classifier, Figure 5.13 allows for the comparison of the three relevance measures in terms of each of the four evaluation measures. For all four evaluation measures the medians and IQRs are similar for the correlation coefficient, IG, and ReliefF. In order to compare the different FS models, consider for example the results based on the correlation coefficient as relevance measure. For all four evaluation measures all of the reduced models perform better than the full model. It can be seen that the model that includes only the highest ranked feature from each feature group exhibits a larger variation than the other models. The results for Hamming-loss and Recall when IG is used as relevance measure are similar with regard to their medians and IQRs, but for Precision and One-error the model Highest does not outperform the full model. This model also exhibits larger variation than the other models. When one considers ReliefF, the reduced models perform better than the full model based on all four evaluations measures, but the model Highest once again does not perform as well as the other reduced models and also exhibits higher variation.

The summaries for all 24 synthetic datasets are available in Appendix E[7]. Once again, the Method of Pairwise Comparisons is applied to aid the analysis of all the procedures and to obtain a ranking of their performances. The IQRs are included to provide insight into the amount of variation that is present in the results. As before, a darker green indicates a larger IQR and the full model is shaded pink.

[7] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

Table 5.14: Hamming-loss for XGBoost.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Highest	IG Highest	RPFS Full
	IQR	0.0176	0.0185	0.0166	0.0188	0.0148	0.0182	0.0265	0.0241	0.0283	0.0176
	Feature Reduction	53		52		46		56	62	64	
Dataset 2	Median	Cor Relevant	Cor Highest 2	IG Relevant	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0144	0.0154	0.0100	0.0127	0.0106	0.0172	0.0206	0.0139	0.0151	0.0212
	Feature Reduction	54		53	49			61		63	62
Dataset 3	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0163	0.0161	0.0159	0.0158	0.0159	0.0188	0.0184	0.0153	0.0234	0.0169
	Feature Reduction	49		50	51		49		51	53	
Dataset 4	Median	Cor Relevant	IG Highest 2	IG Relevant	Cor Highest 2	IG Highest	ReliefF Relevant	Cor Highest	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0121	0.0124	0.0136	0.0119	0.0155	0.0114	0.0146	0.0180	0.0108	0.0345
	Feature Reduction	50		50		56	50	59			68
Dataset 5	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Relevant	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0149	0.0159	0.0162	0.0134	0.0137	0.0189	0.0153	0.0167	0.0200	0.0138
	Feature Reduction	48		51	44			49	38	53	
Dataset 6	Median	Cor Relevant	Cor Highest 2	IG Relevant	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	RPFS Full	ReliefF Highest	IG Highest
	IQR	0.0157	0.0146	0.0148	0.0127	0.0127	0.0136	0.0147	0.0143	0.0184	0.0184
	Feature Reduction	49		45	42			60		58	58
Dataset 7	Median	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0158	0.0157	0.0124	0.0127	0.0154	0.0123	0.0124	0.0139	0.0141	0.0155
	Feature Reduction	48		51		49	49		52	55	
Dataset 8	Median	ReliefF Relevant	Cor Relevant	IG Relevant	IG Highest 2	ReliefF Highest 2	ReliefF Highest	IG Highest	Cor Highest 2	RPFS Full	Cor Highest
	IQR	0.0120	0.0139	0.0112	0.0124	0.0141	0.0139	0.0150	0.0150	0.0117	0.0222
	Feature Reduction	50	50	50			58	54			64
Dataset 9	Median	ReliefF Relevant	ReliefF Highest 2	IG Relevant	Cor Relevant	Cor Highest	Cor Highest 2	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0079	0.0095	0.0067	0.0075	0.0075	0.0075	0.0067	0.0127	0.0214	0.0091
	Feature Reduction	83		83	83	83			84	85	
Dataset 10	Median	Cor Highest	Cor Highest 2	Cor Relevant	IG Highest 2	IG Highest	IG Relevant	ReliefF Relevant	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0333	0.0330	0.0270	0.0290	0.0267	0.0237	0.0314	0.0344	0.0302	0.0293
	Feature Reduction	83		71		79	58	48			78
Dataset 11	Median	IG Relevant	Cor Relevant	ReliefF Relevant	Cor Highest 2	IG Highest 2	Cor Highest	ReliefF Highest 2	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0081	0.0103	0.0085	0.0130	0.0151	0.0168	0.0199	0.0087	0.0197	0.0201
	Feature Reduction	84	84	84			86			88	89
Dataset 12	Median	ReliefF Highest	IG Highest	Cor Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	IG Relevant	Cor Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0451	0.0252	0.0362	0.0246	0.0415	0.0314	0.0249	0.0339	0.0398	0.0379
	Feature Reduction	90	83	73			81	76		86	
Dataset 13	Median	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0063	0.0063	0.0063	0.0066	0.0070	0.0076	0.0076	0.0076	0.0097	0.0063
	Feature Reduction	83	83		83		83	83		86	
Dataset 14	Median	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	Cor Highest	ReliefF Relevant	RPFS Full	IG Highest	ReliefF Highest 2	ReliefF Highest
	IQR	0.0276	0.0319	0.0339	0.0418	0.0408	0.0380	0.0265	0.0437	0.0390	0.0493
	Feature Reduction	81		81		85	82		86		89
Dataset 15	Median	ReliefF Relevant	IG Relevant	Cor Relevant	IG Highest 2	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0069	0.0060	0.0065	0.0063	0.0069	0.0073	0.0104	0.0104	0.0108	0.0063
	Feature Reduction	83	83	83				86	86	86	
Dataset 16	Median	ReliefF Relevant	ReliefF Highest 2	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0209	0.0227	0.0239	0.0211	0.0223	0.0235	0.0270	0.0315	0.0308	0.0257
	Feature Reduction	78			80	81		86	85	83	
Dataset 17	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Highest	Cor Highest	RPFS Full
	IQR	0.0056	0.0063	0.0067	0.0064	0.0053	0.0047	0.0120	0.0144	0.0190	0.0070
	Feature Reduction	83		84		82		84	85	86	
Dataset 18	Median	IG Relevant	ReliefF Highest 2	IG Highest 2	ReliefF Highest	ReliefF Relevant	IG Highest	Cor Highest	Cor Highest 2	Cor Relevant	RPFS Full
	IQR	0.0303	0.0311	0.0317	0.0324	0.0375	0.0422	0.0346	0.0293	0.0291	0.0325
	Feature Reduction	77			73	38	88	83		72	
Dataset 19	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0083	0.0080	0.0052	0.0053	0.0093	0.0080	0.0098	0.0139	0.0182	0.0070
	Feature Reduction	84		84		83	85		85	86	
Dataset 20	Median	IG Relevant	IG Highest	Cor Highest	IG Highest 2	Cor Highest 2	Cor Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0368	0.0297	0.0349	0.0330	0.0315	0.0309	0.0335	0.0343	0.0371	0.0312
	Feature Reduction	77	84	81			72		70	44	
Dataset 21	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Relevant	Cor Highest	Cor Highest 2	IG Highest 2	IG Relevant	IG Highest	RPFS Full
	IQR	0.0057	0.0058	0.0059	0.0056	0.0056	0.0056	0.0053	0.0053	0.0043	0.0050
	Feature Reduction	83		84	83	83			83	84	
Dataset 22	Median	Cor Highest 2	Cor Relevant	IG Relevant	Cor Highest	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0246	0.0240	0.0251	0.0299	0.0314	0.0350	0.0338	0.0270	0.0348	0.0447
	Feature Reduction		81	81	85	76				86	84
Dataset 23	Median	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0058	0.0057	0.0063	0.0061	0.0060	0.0082	0.0210	0.0084	0.0434	0.0692
	Feature Reduction	83		83	83	83		88			92
Dataset 24	Median	Cor Relevant	Cor Highest 2	ReliefF Relevant	RPFS Full	Cor Highest	IG Relevant	IG Highest 2	ReliefF Highest 2	ReliefF Highest	IG Highest
	IQR	0.0214	0.0219	0.0226	0.0218	0.0260	0.0279	0.0282	0.0216	0.0250	0.0338
	Feature Reduction	81		75		85	80			82	84

Table 5.15: One-error for XGBoost.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	IG Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0349	0.0294	0.0387	0.0337	0.0360	0.0355	0.0480	0.0393	0.0524	0.0360
	Feature Reduction	53	52			46		62	56	64	
Dataset 2	Median	IG Relevant	IG Highest 2	ReliefF Relevant	Cor Relevant	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0247	0.0280	0.0265	0.0240	0.0246	0.0307	0.0289	0.0250	0.0302	0.0231
	Feature Reduction	53		49	54			63	61	62	
Dataset 3	Median	Cor Highest	Cor Highest 2	Cor Relevant	IG Relevant	ReliefF Highest 2	ReliefF Relevant	IG Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0320	0.0320	0.0316	0.0294	0.0310	0.0298	0.0286	0.0330	0.0346	0.0383
	Feature Reduction	51		50	49		49		51	53	
Dataset 4	Median	ReliefF Relevant	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	IG Highest	Cor Highest	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0398	0.0390	0.0390	0.0320	0.0333	0.0449	0.0399	0.0514	0.0405	0.0620
	Feature Reduction	50		50	50		56	59			68
Dataset 5	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0329	0.0333	0.0363	0.0264	0.0267	0.0312	0.0384	0.0388	0.0471	0.0342
	Feature Reduction	48		51		44	49		38	53	
Dataset 6	Median	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0278	0.0316	0.0332	0.0237	0.0350	0.0274	0.0336	0.0451	0.0391	0.0270
	Feature Reduction	49		42	45			60	58	58	
Dataset 7	Median	Cor Highest 2	Cor Highest	Cor Relevant	ReliefF Relevant	ReliefF Highest 2	IG Highest 2	IG Relevant	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0376	0.0385	0.0376	0.0339	0.0342	0.0371	0.0362	0.0340	0.0413	0.0438
	Feature Reduction		49	48	49			51	52	55	
Dataset 8	Median	ReliefF Relevant	IG Relevant	ReliefF Highest 2	Cor Relevant	ReliefF Highest	IG Highest 2	Cor Highest 2	IG Highest	RPFS Full	Cor Highest
	IQR	0.0390	0.0349	0.0396	0.0313	0.0448	0.0337	0.0444	0.0360	0.0383	0.0627
	Feature Reduction	50	50		50	58			54		64
Dataset 9	Median	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest 2	ReliefF Relevant	IG Highest 2	IG Relevant	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0103	0.0103	0.0103	0.0139	0.0146	0.0090	0.0095	0.0184	0.0277	0.0164
	Feature Reduction	83	83			83		83	84	85	
Dataset 10	Median	Cor Highest	Cor Highest 2	Cor Relevant	IG Relevant	IG Highest 2	IG Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0520	0.0472	0.0450	0.0521	0.0602	0.0597	0.0494	0.0564	0.0595	0.0466
	Feature Reduction	83		71	58		79	48		78	
Dataset 11	Median	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0202	0.0220	0.0154	0.0196	0.0256	0.0273	0.0340	0.0323	0.0366	0.0159
	Feature Reduction	84	84	84		86			88	89	
Dataset 12	Median	Cor Relevant	IG Relevant	IG Highest 2	IG Highest	ReliefF Relevant	Cor Highest 2	ReliefF Highest 2	Cor Highest	RPFS Full	ReliefF Highest
	IQR	0.0257	0.0257	0.0243	0.0290	0.0297	0.0283	0.0290	0.0271	0.0219	0.0338
	Feature Reduction	73	76		83	86			81		90
Dataset 13	Median	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0091	0.0091	0.0091	0.0117	0.0117	0.0117	0.0085	0.0091	0.0122	0.0195
	Feature Reduction	83	83		83	83		83		86	
Dataset 14	Median	Cor Relevant	Cor Highest 2	IG Relevant	Cor Highest	IG Highest 2	IG Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0738	0.0888	0.0773	0.0787	0.0730	0.0732	0.0810	0.0898	0.0850	0.0777
	Feature Reduction	81		81	85		86	82		89	
Dataset 15	Median	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0154	0.0113	0.0135	0.0178	0.0114	0.0165	0.0137	0.0175	0.0193	0.0169
	Feature Reduction	83	83			83		86	86	86	
Dataset 16	Median	Cor Relevant	Cor Highest 2	IG Highest 2	IG Relevant	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0587	0.0611	0.0659	0.0678	0.0419	0.0597	0.0446	0.0713	0.0491	0.0517
	Feature Reduction	81			80	78	86		85	83	
Dataset 17	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	IG Highest	Cor Highest	RPFS Full
	IQR	0.0147	0.0146	0.0170	0.0180	0.0123	0.0108	0.0236	0.0251	0.0328	0.0171
	Feature Reduction	83		84			82	84	85	86	
Dataset 18	Median	Cor Highest	Cor Highest 2	Cor Relevant	RPFS Full	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest	ReliefF Relevant	ReliefF Highest 2
	IQR	0.0566	0.0584	0.0521	0.0479	0.0623	0.0496	0.0803	0.0627	0.0623	0.0594
	Feature Reduction	83		72			77	88	73	38	
Dataset 19	Median	Cor Highest 2	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0255	0.0252	0.0261	0.0245	0.0256	0.0269	0.0288	0.0280	0.0346	0.0233
	Feature Reduction		84	84		85	83		85	86	
Dataset 20	Median	Cor Highest	Cor Relevant	Cor Highest 2	IG Highest	IG Highest 2	IG Relevant	ReliefF Relevant	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0423	0.0413	0.0397	0.0354	0.0402	0.0394	0.0370	0.0345	0.0320	0.0317
	Feature Reduction	81	72		84		77	44			70
Dataset 21	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	Cor Highest	Cor Highest 2	RPFS Full
	IQR	0.0131	0.0131	0.0129	0.0153	0.0152	0.0148	0.0135	0.0135	0.0135	0.0160
	Feature Reduction	83		84	83		84	83	83		
Dataset 22	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	ReliefF Relevant	ReliefF Highest 2	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0849	0.0842	0.0873	0.0947	0.0854	0.0832	0.0822	0.0851	0.0846	0.0816
	Feature Reduction	81		85		81	76			86	84
Dataset 23	Median	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0135	0.0135	0.0135	0.0144	0.0117	0.0206	0.0309	0.0212	0.0550	0.0857
	Feature Reduction	83	83		83	83		88			92
Dataset 24	Median	Cor Relevant	Cor Highest 2	IG Relevant	ReliefF Relevant	Cor Highest	IG Highest 2	ReliefF Highest 2	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0582	0.0566	0.0574	0.0562	0.0580	0.0545	0.0579	0.0622	0.0609	0.0577
	Feature Reduction	81		80	75	85				84	82

Table 5.16: Precision for XGBoost.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	IG Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	IG Highest	RPFS Full	ReliefF Highest
	IQR	0.0334	0.0327	0.0398	0.0382	0.0316	0.0336	0.0421	0.0434	0.0350	0.0388
	Feature Reduction	53	52			46		62	64		56
Dataset 2	Median	IG Relevant	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0370	0.0415	0.0395	0.0415	0.0406	0.0446	0.0417	0.0422	0.0443	0.0465
	Feature Reduction	53	54		49			61		63	62
Dataset 3	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0244	0.0257	0.0306	0.0290	0.0313	0.0227	0.0223	0.0192	0.0343	0.0274
	Feature Reduction	49		50	51		49		51	53	
Dataset 4	Median	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Highest 2	IG Relevant	IG Highest	Cor Highest	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0366	0.0348	0.0316	0.0414	0.0406	0.0420	0.0374	0.0525	0.0343	0.0737
	Feature Reduction	50		50		50	56	59			68
Dataset 5	Median	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	IG Highest 2	IG Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0415	0.0418	0.0454	0.0461	0.0428	0.0464	0.0478	0.0399	0.0532	0.0493
	Feature Reduction	48		51	38		44		49	53	
Dataset 6	Median	Cor Relevant	Cor Highest 2	IG Relevant	ReliefF Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	RPFS Full	ReliefF Highest	IG Highest
	IQR	0.0467	0.0494	0.0469	0.0471	0.0417	0.0547	0.0481	0.0486	0.0490	0.0542
	Feature Reduction	49		45	42			60		58	58
Dataset 7	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Highest 2	Cor Relevant	Cor Highest	IG Highest 2	RPFS Full	IG Relevant	IG Highest
	IQR	0.0249	0.0249	0.0266	0.0258	0.0265	0.0275	0.0265	0.0300	0.0266	0.0309
	Feature Reduction	49		52		48	49			51	55
Dataset 8	Median	Cor Relevant	IG Relevant	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	RPFS Full	Cor Highest
	IQR	0.0363	0.0344	0.0318	0.0318	0.0348	0.0374	0.0309	0.0369	0.0398	0.0594
	Feature Reduction	50	50	50				54	58		64
Dataset 9	Median	IG Highest 2	IG Relevant	ReliefF Highest 2	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0135	0.0154	0.0162	0.0149	0.0158	0.0158	0.0160	0.0249	0.0301	0.0198
	Feature Reduction		83		83	83		83	84	85	
Dataset 10	Median	Cor Highest	Cor Relevant	Cor Highest 2	IG Highest 2	IG Highest	IG Relevant	ReliefF Relevant	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0543	0.0470	0.0496	0.0540	0.0477	0.0428	0.0463	0.0626	0.0512	0.0439
	Feature Reduction	83	71			79	58	48			78
Dataset 11	Median	IG Relevant	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Highest 2	RPFS Full	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest
	IQR	0.0267	0.0211	0.0227	0.0251	0.0352	0.0252	0.0383	0.0336	0.0373	0.0416
	Feature Reduction	84	84		84			86		88	89
Dataset 12	Median	IG Highest	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	Cor Relevant	Cor Highest 2	Cor Highest	RPFS Full
	IQR	0.0729	0.0640	0.0738	0.0688	0.0612	0.0710	0.0792	0.0849	0.0842	0.0852
	Feature Reduction	83	86		76		90	73		81	
Dataset 13	Median	ReliefF Relevant	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0089	0.0090	0.0090	0.0090	0.0088	0.0111	0.0111	0.0111	0.0109	0.0112
	Feature Reduction	83	83	83			83	83		86	
Dataset 14	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Highest	ReliefF Relevant	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0645	0.0631	0.0606	0.0707	0.0805	0.0948	0.0773	0.0545	0.0719	0.0934
	Feature Reduction	81	81			85	86	82			89
Dataset 15	Median	IG Relevant	ReliefF Relevant	Cor Relevant	IG Highest 2	Cor Highest 2	ReliefF Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0124	0.0133	0.0123	0.0149	0.0124	0.0168	0.0229	0.0235	0.0229	0.0160
	Feature Reduction	83	83	83				86	86	86	
Dataset 16	Median	IG Highest 2	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0881	0.0861	0.0653	0.0942	0.0802	0.0691	0.0985	0.0745	0.0864	0.0839
	Feature Reduction		80	78	81			85	83	86	
Dataset 17	Median	IG Relevant	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest	IG Highest	Cor Highest	RPFS Full
	IQR	0.0129	0.0190	0.0140	0.0186	0.0180	0.0187	0.0312	0.0365	0.0418	0.0172
	Feature Reduction	83	82			84		84	85	86	
Dataset 18	Median	Cor Highest 2	Cor Highest	Cor Relevant	RPFS Full	IG Relevant	ReliefF Relevant	IG Highest 2	IG Highest	ReliefF Highest	ReliefF Highest 2
	IQR	0.0584	0.0525	0.0490	0.0497	0.0829	0.0855	0.0794	0.0961	0.0834	0.0828
	Feature Reduction		83	72		77	38		88	73	
Dataset 19	Median	Cor Relevant	Cor Highest 2	IG Relevant	Cor Highest	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	IG Highest	RPFS Full	ReliefF Highest
	IQR	0.0252	0.0248	0.0222	0.0296	0.0283	0.0233	0.0329	0.0322	0.0287	0.0454
	Feature Reduction	84		84	85	83			85		86
Dataset 20	Median	Cor Highest	IG Highest	Cor Highest 2	IG Relevant	IG Highest 2	Cor Relevant	ReliefF Relevant	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0953	0.0876	0.0998	0.0848	0.0868	0.0989	0.0944	0.0821	0.0821	0.0890
	Feature Reduction	81	84		77		72	44			70
Dataset 21	Median	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0074	0.0074	0.0078	0.0078	0.0073	0.0092	0.0092	0.0092	0.0077	0.0104
	Feature Reduction	83		83		84	83	83		84	
Dataset 22	Median	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	Cor Highest	ReliefF Relevant	RPFS Full	ReliefF Highest 2	IG Highest	ReliefF Highest
	IQR	0.0691	0.0688	0.0691	0.0742	0.0756	0.0817	0.0869	0.0948	0.0955	0.1004
	Feature Reduction	81		81		85	76			86	84
Dataset 23	Median	IG Relevant	IG Highest	IG Highest 2	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0153	0.0153	0.0153	0.0146	0.0197	0.0203	0.0459	0.0235	0.0962	0.1399
	Feature Reduction	83	83		83	83		88			92
Dataset 24	Median	Cor Relevant	Cor Highest 2	ReliefF Highest 2	ReliefF Relevant	RPFS Full	Cor Highest	IG Relevant	ReliefF Highest	IG Highest 2	IG Highest
	IQR	0.0719	0.0753	0.0644	0.0633	0.0622	0.0756	0.0709	0.0815	0.0761	0.0877
	Feature Reduction	81			75		85	80	82		84

Table 5.17: Recall for XGBoost.

	Rank	1	2	3	4	5	6	7	8	9	10
Dataset 1	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0528	0.0451	0.0522	0.0476	0.0500	0.0520	0.0572	0.0541	0.0533	0.0526
	Feature Reduction	52	53			46		62	56	64	
Dataset 2	Median	ReliefF Relevant	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0432	0.0535	0.0543	0.0531	0.0544	0.0484	0.0440	0.0466	0.0526	0.0503
	Feature Reduction	49	54		53			61	62	63	
Dataset 3	Median	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	IG Relevant	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0419	0.0424	0.0413	0.0425	0.0394	0.0383	0.0446	0.0411	0.0476	0.0437
	Feature Reduction	50	51		49	49			51	53	
Dataset 4	Median	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	RPFS Full	ReliefF Highest
	IQR	0.0524	0.0463	0.0493	0.0553	0.0435	0.0413	0.0512	0.0529	0.0460	0.0709
	Feature Reduction	50		50	56	50		59			68
Dataset 5	Median	Cor Highest 2	Cor Relevant	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0550	0.0550	0.0561	0.0551	0.0541	0.0534	0.0488	0.0450	0.0485	0.0575
	Feature Reduction		48	51		44	49		53	38	
Dataset 6	Median	Cor Relevant	IG Relevant	ReliefF Relevant	IG Highest 2	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0596	0.0435	0.0551	0.0401	0.0624	0.0614	0.0610	0.0547	0.0568	0.0632
	Feature Reduction	49	45	42			60		58	58	
Dataset 7	Median	IG Highest 2	IG Relevant	ReliefF Relevant	ReliefF Highest 2	IG Highest	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0356	0.0380	0.0427	0.0430	0.0399	0.0402	0.0402	0.0402	0.0404	0.0428
	Feature Reduction		51	49		55	48		49	52	
Dataset 8	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest 2	RPFS Full	Cor Highest
	IQR	0.0548	0.0570	0.0567	0.0510	0.0524	0.0534	0.0536	0.0602	0.0604	0.0909
	Feature Reduction	50		58	50		50	54			64
Dataset 9	Median	ReliefF Highest 2	ReliefF Relevant	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest	IG Highest 2	IG Relevant	IG Highest	RPFS Full
	IQR	0.0188	0.0189	0.0239	0.0237	0.0237	0.0250	0.0222	0.0220	0.0254	0.0253
	Feature Reduction		83	83	83		84		83	85	
Dataset 10	Median	Cor Highest	Cor Highest 2	Cor Relevant	IG Highest	ReliefF Relevant	ReliefF Highest 2	RPFS Full	IG Highest 2	ReliefF Highest	IG Relevant
	IQR	0.0641	0.0760	0.0608	0.0779	0.0805	0.1066	0.0583	0.0834	0.0897	0.0823
	Feature Reduction	83		71	79	48				78	58
Dataset 11	Median	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0260	0.0278	0.0302	0.0327	0.0305	0.0310	0.0429	0.0409	0.0463	0.0321
	Feature Reduction	84	84	84			86		88	89	
Dataset 12	Median	IG Highest	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	RPFS Full	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest
	IQR	0.0853	0.0916	0.0969	0.0764	0.0790	0.0892	0.0774	0.0995	0.0944	0.0874
	Feature Reduction	83		76	73			81		86	90
Dataset 13	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0169	0.0188	0.0133	0.0133	0.0133	0.0130	0.0130	0.0130	0.0204	0.0164
	Feature Reduction	83		83	83		83	83		86	
Dataset 14	Median	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	RPFS Full	IG Highest	ReliefF Highest
	IQR	0.0749	0.0797	0.0722	0.0934	0.0554	0.1010	0.0772	0.0611	0.0894	0.0873
	Feature Reduction	81		81		82	85			86	89
Dataset 15	Median	ReliefF Relevant	IG Relevant	ReliefF Highest 2	IG Highest 2	Cor Relevant	IG Highest	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0192	0.0213	0.0241	0.0227	0.0298	0.0275	0.0308	0.0319	0.0365	0.0269
	Feature Reduction	83	83			83	86		86	86	
Dataset 16	Median	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest	IG Relevant	IG Highest 2	RPFS Full	Cor Highest	IG Highest
	IQR	0.0645	0.0669	0.0743	0.0799	0.0721	0.0789	0.0816	0.0730	0.0849	0.0989
	Feature Reduction	78		81		83	80			86	85
Dataset 17	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	IG Highest	ReliefF Relevant	ReliefF Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0203	0.0212	0.0224	0.0219	0.0252	0.0296	0.0288	0.0427	0.0257	0.0277
	Feature Reduction	83		84		85	82		86	84	
Dataset 18	Median	Cor Highest 2	Cor Highest	Cor Relevant	RPFS Full	IG Highest 2	IG Highest	IG Relevant	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest
	IQR	0.0885	0.0789	0.0851	0.0687	0.0983	0.0956	0.1066	0.0876	0.0969	0.0881
	Feature Reduction		83	72			88	77	38		73
Dataset 19	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0332	0.0339	0.0384	0.0375	0.0337	0.0400	0.0333	0.0343	0.0522	0.0408
	Feature Reduction	84		84		83	85		85	86	
Dataset 20	Median	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest	IG Relevant	ReliefF Highest 2	IG Highest	IG Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0835	0.0874	0.1084	0.0868	0.0795	0.0920	0.0762	0.0902	0.0932	0.1089
	Feature Reduction	72		81	70	77		84		44	
Dataset 21	Median	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	Cor Relevant	Cor Highest	Cor Highest 2	RPFS Full
	IQR	0.0213	0.0213	0.0190	0.0190	0.0213	0.0185	0.0173	0.0173	0.0173	0.0178
	Feature Reduction	83		83		84	84	83	83		
Dataset 22	Median	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Relevant	Cor Highest	ReliefF Highest 2	RPFS Full	IG Highest 2	ReliefF Highest	IG Highest
	IQR	0.0903	0.0972	0.0988	0.1014	0.1162	0.1062	0.1032	0.1282	0.1309	0.1199
	Feature Reduction	81		76	81	85				84	86
Dataset 23	Median	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	RPFS Full	ReliefF Highest 2	ReliefF Highest
	IQR	0.0227	0.0221	0.0228	0.0217	0.0217	0.0250	0.0494	0.0229	0.1160	0.1763
	Feature Reduction	83		83	83	83		88			92
Dataset 24	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	IG Highest	RPFS Full	ReliefF Relevant	Cor Highest	ReliefF Highest 2	ReliefF Highest
	IQR	0.1281	0.1194	0.0970	0.0977	0.1058	0.1031	0.0834	0.0886	0.0798	0.0809
	Feature Reduction	80		81		84		75	85		82

The ranked performances of the procedures using XGBoost as classifier, as summarised in Tables 5.14 to 5.17, do not exhibit the same pattern as for the SVM classifier. The procedures based on the reduced models do not consistently outperform the full model for all the datasets. It is also not as easy as in the SVM case to determine which procedures perform well across all datasets and evaluation measures. This can probably be ascribed to the way that the

thresholding is applied. The thresholds are determined based on the SVM classifier, and these thresholds are then applied to the XGBoost classifier.

For example, for Hamming-loss, the FS procedure ReliefF Relevant ranks first for Datasets 8, 9, 12, 15, 16, and 21. For Dataset 21, all three of the procedures based on ReliefF are ranked higher than the procedures based on the correlation coefficient and IG. For One-error and Precision, ReliefF Relevant ranks highest for Dataset 4 and Dataset 3, respectively. Finally, from Table 5.17 for Recall, one can see that ReliefF Relevant ranks first for Datasets 8, 9, 13, 15, 16 and 21.

Applying the Method of Pairwise Comparisons using the median (over all 24 datasets) of the IQR as a measure of variation results in Table 5.18. This table provides a ranking of the methods over all 24 datasets and provides a global summary. Values shaded in darker green represent higher variation.

Table 5.18: Method of Pairwise Comparisons for all 24 datasets – XGBoost.

	Hamming Loss		One-Error		Precision		Recall	
Rank	Technique	Median IQR	Technique	Median IQR	Technique	Median IQR	Technique	Median IQR
1	Cor Relevant	0.01528	Cor Relevant	0.03034	Cor Relevant	0.03646	Cor Relevant	0.05308
2	IG Relevant	0.01350	Cor Highest 2	0.03268	Cor Highest 2	0.03847	IG Relevant	0.04806
3	Cor Highest 2	0.01553	IG Relevant	0.02805	IG Relevant	0.03570	Cor Highest 2	0.05325
4	IG Highest 2	0.01318	IG Highest 2	0.02831	ReliefF Relevant	0.03169	ReliefF Relevant	0.04891
5	ReliefF Relevant	0.01274	ReliefF Relevant	0.03354	IG Highest 2	0.03939	IG Highest 2	0.04695
6	Cor Highest	0.01980	Cor Highest	0.03494	ReliefF Highest 2	0.04315	Cor Highest	0.05365
7	ReliefF Highest 2	0.01825	ReliefF Highest 2	0.03463	Cor Highest	0.04375	ReliefF Highest 2	0.05247
8	IG Highest	0.01692	IG Highest	0.03419	IG Highest	0.04092	IG Highest	0.05334
9	ReliefF Highest	0.02231	ReliefF Highest	0.03786	RPFS Full	0.03739	ReliefF Highest	0.05442
10	RPFS Full	0.01410	RPFS Full	0.03438	ReliefF Highest	0.04596	RPFS Full	0.05143

The procedures Correlation Relevant, Correlation Highest 2, and IG Relevant outperform their counterparts on all four evaluation measures. From Table 5.18 one can also see that the variation associated with these four procedures are lower than for example the variation in the results for ReliefF Highest, which is ranked last of the FS procedures. ReliefF Highest also performs worse than the full model for Precision.

As in the case of the SVM classifier, one also wishes to assess the procedures based on the properties associated with multi-label datasets. The datasets are therefore grouped according to the signal strength, the number of irrelevant features, the number of training instances, the

label dependence, and the vector of label densities. Tables 5.19 to 5.24 summarise the ranked medians (over all of the datasets) of the procedures using the Method of Pairwise Comparisons.

Based on Table 5.19, one can conclude that the ranking of ReliefF Relevant improves as the signal strength increases from 10 to 100. This is the same result that was observed for the SVM classifier. However, although the procedure Correlation Relevant ranks high for all other cases, its ranking drops to fourth when considering Recall as evaluation measure and when the signal strength increases.

For Table 5.20, where the datasets are grouped according to the number of irrelevant features, it is somewhat surprising that the level of noise does not seem to influence the ranking of the procedures.

Tables 5.21 and 5.22 rank the procedures for the cases corresponding to different numbers of training instances. The procedures based on the correlation coefficient rank higher for smaller training datasets, $N = 30$, irrespective of evaluation measure. This result was also observed for the SVM classifier. It is interesting to note that for One-error, the ranking for IG Relevant improves as the number of training instances increases. For $N = 30$, IG Relevant ranks fifth, for $N = 80$, the ranking of IG Relevant improves to third, and the procedure ranks first when $N = 240$. The same pattern is observed for the ranking of ReliefF Relevant for Recall. The ranking of the procedure improves as the number of training instances increases.

From Table 5.23 it can be seen that the procedures are ranked fairly consistently for Hamming-loss, One-error, and Recall. The ranks of the FS methods IG Relevant and IG Highest 2 improve when the labels are correlated. For Precision, the ranking of IG Relevant improves substantially when correlation is present. The procedure is ranked fourth when there is no label dependence but ranks first when the labels are correlated.

In Table 5.24 the datasets are grouped based on the vector of densities. For Hamming-loss, One-error, and Precision, the rankings remain fairly consistent. However, for Recall, the FS procedures IG Relevant and IG Highest 2 rank higher when the vector of densities varies. The opposite is true for the FS procedure ReliefF Relevant.

Table 5.19: Method of Pairwise Comparisons for the signal level – XGBoost.

Rank	Hamming Loss				One-Error				Precision				Recall			
	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR
1	IG Relevant	0.01413	Cor Relevant	0.01485	Cor Relevant	0.02679	Cor Relevant	0.03178	IG Relevant	0.03989	Cor Relevant	0.03347	Cor Relevant	0.05424	ReliefF Relevant	0.04598
2	Cor Relevant	0.01528	Cor Highest 2	0.01532	Cor Highest 2	0.02993	Cor Highest 2	0.03545	Cor Relevant	0.04152	ReliefF Relevant	0.02826	Cor Highest 2	0.05464	IG Relevant	0.04523
3	Cor Highest 2	0.01568	IG Relevant	0.01296	IG Relevant	0.02588	IG Relevant	0.03556	Cor Highest 2	0.04080	IG Relevant	0.03051	IG Relevant	0.04908	IG Highest 2	0.04231
4	IG Highest 2	0.01437	ReliefF Relevant	0.01217	IG Highest 2	0.02735	ReliefF Relevant	0.03645	ReliefF Relevant	0.04384	Cor Highest 2	0.03302	IG Highest 2	0.05097	Cor Relevant	0.04266
5	ReliefF Relevant	0.01379	IG Highest 2	0.01257	Cor Highest	0.03317	IG Highest 2	0.03540	IG Highest 2	0.04171	IG Highest 2	0.02918	ReliefF Relevant	0.04927	ReliefF Highest 2	0.05495
6	ReliefF Highest 2	0.01852	Cor Highest	0.01842	ReliefF Relevant	0.03141	Cor Highest	0.03923	Cor Highest	0.04375	ReliefF Highest 2	0.04365	Cor Highest	0.05666	Cor Highest 2	0.04131
7	Cor Highest	0.01980	ReliefF Highest 2	0.01818	ReliefF Highest 2	0.03297	ReliefF Highest 2	0.04210	ReliefF Highest 2	0.04315	Cor Highest	0.04169	ReliefF Highest 2	0.05039	IG Highest	0.04735
8	IG Highest	0.02051	IG Highest	0.01517	IG Highest	0.03174	IG Highest	0.03866	IG Highest	0.04388	IG Highest	0.03090	IG Highest	0.05334	Cor Highest	0.05030
9	ReliefF Highest	0.02064	ReliefF Highest	0.02424	ReliefF Highest	0.03561	ReliefF Highest	0.04693	RPFS Full	0.04543	ReliefF Highest	0.05529	ReliefF Highest	0.05317	ReliefF Highest	0.06378
10	RPFS Full	0.01410	RPFS Full	0.01361	RPFS Full	0.02514	RPFS Full	0.03941	ReliefF Highest	0.04596	RPFS Full	0.03216	RPFS Full	0.05502	RPFS Full	0.04487

Table 5.20: Method of Pairwise Comparisons for the number of irrelevant features – XGBoost.

	Hamming Loss				One-Error				Precision				Recall			
Rank	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR
1	Cor Relevant	0.01528	IG Relevant	0.01469	Cor Relevant	0.03143	Cor Relevant	0.02545	Cor Relevant	0.03646	IG Relevant	0.03474	Cor Relevant	0.05308	Cor Relevant	0.04962
2	IG Relevant	0.01350	Cor Relevant	0.01589	IG Relevant	0.03214	Cor Highest 2	0.02686	Cor Highest 2	0.03847	Cor Relevant	0.03607	IG Relevant	0.04806	IG Relevant	0.05271
3	Cor Highest 2	0.01553	Cor Highest 2	0.01747	Cor Highest 2	0.03334	IG Relevant	0.02588	ReliefF Relevant	0.03169	Cor Highest 2	0.03724	ReliefF Relevant	0.04891	Cor Highest 2	0.05675
4	IG Highest 2	0.01268	IG Highest 2	0.01949	ReliefF Relevant	0.03496	Cor Highest	0.03183	IG Relevant	0.03570	ReliefF Relevant	0.03731	IG Highest 2	0.04695	ReliefF Relevant	0.04457
5	ReliefF Relevant	0.01274	ReliefF Relevant	0.01513	IG Highest 2	0.03114	IG Highest 2	0.02586	IG Highest 2	0.03939	IG Highest 2	0.04461	Cor Highest 2	0.05325	IG Highest 2	0.05779
6	Cor Highest	0.01599	Cor Highest	0.02352	ReliefF Highest 2	0.03528	ReliefF Relevant	0.02829	ReliefF Highest 2	0.03826	Cor Highest	0.04921	Cor Highest	0.05365	Cor Highest	0.05674
7	ReliefF Highest 2	0.01758	ReliefF Highest 2	0.02213	Cor Highest	0.03741	ReliefF Highest 2	0.03297	Cor Highest	0.04186	ReliefF Highest 2	0.06347	ReliefF Highest 2	0.05039	ReliefF Highest 2	0.07207
8	IG Highest	0.01530	IG Highest	0.02327	IG Highest	0.03866	IG Highest	0.03063	IG Highest	0.04092	IG Highest	0.04251	IG Highest	0.05334	IG Highest	0.05853
9	ReliefF Highest	0.02060	ReliefF Highest	0.02716	ReliefF Highest	0.03918	ReliefF Highest	0.03561	RPFS Full	0.03739	RPFS Full	0.03920	ReliefF Highest	0.05086	ReliefF Highest	0.07650
10	RPFS Full	0.01410	RPFS Full	0.01544	RPFS Full	0.03713	RPFS Full	0.02261	ReliefF Highest	0.04267	ReliefF Highest	0.05820	RPFS Full	0.05143	RPFS Full	0.04952

Table 5.21: Method of Pairwise Comparisons for the number of training instances: Hamming-loss and One-error – XGBoost.

	Hamming Loss						One-Error					
Rank	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR
1	Cor Relevant	0.02731	Cor Relevant	0.01528	IG Relevant	0.00633	Cor Relevant	0.05511	Cor Relevant	0.03143	IG Relevant	0.01406
2	IG Relevant	0.02653	IG Relevant	0.01350	Cor Relevant	0.00640	Cor Highest 2	0.05748	IG Relevant	0.03214	Cor Relevant	0.01486
3	Cor Highest 2	0.03039	Cor Highest 2	0.01553	ReliefF Relevant	0.00675	Cor Highest	0.05729	Cor Highest 2	0.03334	Cor Highest 2	0.01785
4	Cor Highest	0.03234	IG Highest 2	0.01268	Cor Highest 2	0.00663	IG Relevant	0.05475	ReliefF Relevant	0.03496	IG Highest 2	0.01408
5	IG Highest 2	0.03037	ReliefF Relevant	0.01274	IG Highest 2	0.00650	IG Highest 2	0.06125	IG Highest 2	0.03114	ReliefF Relevant	0.01241
6	ReliefF Relevant	0.03425	Cor Highest	0.01599	Cor Highest	0.00917	IG Highest	0.06606	ReliefF Highest 2	0.03528	Cor Highest	0.02154
7	IG Highest	0.03265	ReliefF Highest 2	0.01758	ReliefF Highest 2	0.00839	ReliefF Relevant	0.05280	Cor Highest	0.03741	ReliefF Highest 2	0.01519
8	ReliefF Highest 2	0.03362	IG Highest	0.01530	IG Highest	0.01214	ReliefF Highest 2	0.05714	IG Highest	0.03866	IG Highest	0.01993
9	RPFS Full	0.02860	ReliefF Highest	0.02060	ReliefF Highest	0.01236	RPFS Full	0.04978	ReliefF Highest	0.03918	ReliefF Highest	0.02148
10	ReliefF Highest	0.03337	RPFS Full	0.01410	RPFS Full	0.00700	ReliefF Highest	0.05855	RPFS Full	0.03713	RPFS Full	0.01703

Table 5.22: Method of Pairwise Comparisons for the number of training instances: Precision and Recall – XGBoost.

	Precision						Recall					
Rank	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR
1	Cor Relevant	0.07051	Cor Relevant	0.03646	IG Relevant	0.01407	Cor Relevant	0.07996	Cor Relevant	0.05308	ReliefF Relevant	0.02148
2	Cor Highest 2	0.07204	Cor Highest 2	0.03847	ReliefF Relevant	0.01477	Cor Highest 2	0.08368	IG Relevant	0.04806	IG Relevant	0.02164
3	IG Relevant	0.06999	ReliefF Relevant	0.03169	IG Highest 2	0.01447	IG Relevant	0.08964	ReliefF Relevant	0.04891	IG Highest 2	0.02215
4	Cor Highest	0.07808	IG Relevant	0.03570	Cor Relevant	0.01688	Cor Highest	0.08674	IG Highest 2	0.04695	Cor Relevant	0.02316
5	IG Highest 2	0.07511	IG Highest 2	0.03939	Cor Highest 2	0.01724	IG Highest 2	0.09250	Cor Highest 2	0.05325	ReliefF Highest 2	0.02646
6	IG Highest	0.09122	ReliefF Highest 2	0.03826	Cor Highest	0.02625	ReliefF Relevant	0.08548	Cor Highest	0.05365	Cor Highest 2	0.02435
7	ReliefF Relevant	0.07129	Cor Highest	0.04186	ReliefF Highest 2	0.01771	IG Highest	0.09249	ReliefF Highest 2	0.05039	Cor Highest	0.03375
8	RPFS Full	0.07215	IG Highest	0.04092	IG Highest	0.02651	ReliefF Highest 2	0.09444	IG Highest	0.05334	IG Highest	0.02529
9	ReliefF Highest 2	0.07287	RPFS Full	0.03739	ReliefF Highest	0.02807	RPFS Full	0.08109	ReliefF Highest	0.05086	ReliefF Highest	0.02880
10	ReliefF Highest	0.08243	ReliefF Highest	0.04267	RPFS Full	0.01850	ReliefF Highest	0.08737	RPFS Full	0.05143	RPFS Full	0.02607

Table 5.23: Method of Pairwise Comparisons for the label dependence – XGBoost.

	Hamming Loss				One-Error				Precision				Recall			
Rank	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR
1	Cor Relevant	0.01584	Cor Relevant	0.01416	Cor Highest 2	0.03546	Cor Relevant	0.02679	Cor Relevant	0.03162	IG Relevant	0.03882	Cor Highest 2	0.04675	IG Relevant	0.05169
2	Cor Highest 2	0.01593	IG Relevant	0.01237	Cor Relevant	0.03223	IG Relevant	0.02588	Cor Highest 2	0.03551	Cor Relevant	0.03904	Cor Relevant	0.04734	IG Highest 2	0.04937
3	IG Relevant	0.01487	Cor Highest 2	0.01478	Cor Highest	0.03741	IG Highest 2	0.02771	ReliefF Relevant	0.02827	Cor Highest 2	0.03847	ReliefF Relevant	0.04558	ReliefF Relevant	0.05206
4	Cor Highest	0.01760	IG Highest 2	0.01256	IG Relevant	0.03214	Cor Highest 2	0.02993	IG Relevant	0.03000	ReliefF Relevant	0.03666	Cor Highest	0.04941	Cor Relevant	0.05342
5	IG Highest 2	0.01489	ReliefF Relevant	0.01239	IG Highest 2	0.03114	ReliefF Relevant	0.03141	IG Highest 2	0.03236	IG Highest 2	0.04101	IG Relevant	0.04226	Cor Highest 2	0.05725
6	ReliefF Relevant	0.01578	ReliefF Highest 2	0.01894	ReliefF Relevant	0.03496	Cor Highest	0.03222	Cor Highest	0.04194	ReliefF Highest 2	0.04855	IG Highest 2	0.04294	Cor Highest	0.05628
7	ReliefF Highest 2	0.01825	Cor Highest	0.02082	ReliefF Highest 2	0.03488	ReliefF Highest 2	0.03449	ReliefF Highest 2	0.02966	Cor Highest	0.04704	ReliefF Highest 2	0.04666	ReliefF Highest 2	0.05899
8	IG Highest	0.01835	IG Highest	0.01692	IG Highest	0.03715	IG Highest	0.03383	IG Highest	0.03820	IG Highest	0.04315	IG Highest	0.04719	IG Highest	0.05450
9	ReliefF Highest	0.02171	RPFS Full	0.01284	ReliefF Highest	0.03691	ReliefF Highest	0.03786	ReliefF Highest	0.03657	RPFS Full	0.04102	ReliefF Highest	0.04628	ReliefF Highest	0.06378
10	RPFS Full	0.01620	ReliefF Highest	0.02312	RPFS Full	0.03713	RPFS Full	0.02514	RPFS Full	0.03247	ReliefF Highest	0.05997	RPFS Full	0.04815	RPFS Full	0.05531

Table 5.24: Method of Pairwise Comparisons for different vectors of density – XGBoost.

Rank	Hamming Loss				One-Error				Precision				Recall			
	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR
1	Cor Relevant	0.01515	Cor Relevant	0.01528	Cor Relevant	0.02758	Cor Relevant	0.03205	IG Relevant	0.03519	Cor Relevant	0.03894	ReliefF Relevant	0.04627	IG Relevant	0.04725
2	IG Relevant	0.01493	IG Relevant	0.01290	Cor Highest 2	0.03014	Cor Highest 2	0.03546	Cor Relevant	0.03462	Cor Highest 2	0.03961	Cor Relevant	0.04812	IG Highest 2	0.04627
3	Cor Highest 2	0.01566	Cor Highest 2	0.01532	IG Relevant	0.02751	IG Relevant	0.03081	ReliefF Relevant	0.03158	IG Relevant	0.04040	Cor Highest 2	0.04678	Cor Relevant	0.05417
4	IG Highest 2	0.01559	IG Highest 2	0.01268	IG Highest 2	0.02831	IG Highest 2	0.03056	Cor Highest 2	0.03717	ReliefF Relevant	0.03897	IG Relevant	0.04872	Cor Highest 2	0.05760
5	ReliefF Relevant	0.01379	ReliefF Relevant	0.01252	ReliefF Relevant	0.02973	Cor Highest	0.03741	IG Highest 2	0.03939	IG Highest 2	0.03734	IG Highest 2	0.04695	ReliefF Relevant	0.05166
6	Cor Highest	0.01868	Cor Highest	0.02002	Cor Highest	0.02956	ReliefF Relevant	0.03544	ReliefF Highest 2	0.03913	Cor Highest	0.04704	Cor Highest	0.04760	Cor Highest	0.05873
7	ReliefF Highest 2	0.01825	ReliefF Highest 2	0.01647	ReliefF Highest 2	0.03249	ReliefF Highest 2	0.03670	Cor Highest	0.03998	ReliefF Highest 2	0.04472	ReliefF Highest 2	0.05021	IG Highest	0.05351
8	IG Highest	0.02051	ReliefF Highest	0.01921	IG Highest	0.03264	IG Highest	0.03570	IG Highest	0.04268	IG Highest	0.03820	IG Highest	0.05296	ReliefF Highest 2	0.05899
9	ReliefF Highest	0.02494	IG Highest	0.01514	ReliefF Highest	0.03560	ReliefF Highest	0.04194	RPFS Full	0.03466	RPFS Full	0.04423	ReliefF Highest	0.05086	ReliefF Highest	0.05569
10	RPFS Full	0.01543	RPFS Full	0.01405	RPFS Full	0.02951	RPFS Full	0.03438	ReliefF Highest	0.04278	ReliefF Highest	0.05109	RPFS Full	0.04815	RPFS Full	0.05890

Feature Reduction

The FS procedures based on selecting all the features that were identified as relevant, as well as those selecting only the highest ranked feature from each group of relevant features, are compared for all 24 datasets in Figures 5.14 to 5.16.

Datasets 1 to 8 each contain ten relevant and ten irrelevant features. One would expect lower values of Feature Reduction for datasets where the proportion of irrelevant features is low relative to datasets with a higher proportion of irrelevant features. The results presented in Figure 5.14 confirms this expectation when compared with Figures 5.15 and 5.16 – as was the case for the SVM classifier. Most values of the Feature Reduction lie between 40 and 60.

On the other hand, one would expect higher values of Feature Reduction for datasets that contain a higher proportion of irrelevant features. Datasets 9 to 24 each contain ten relevant and 50 irrelevant features. Higher Feature Reduction values of approximately 80 are observed in Figures 5.15 and 5.16. As was the case for the SVM classifier, the results for Datasets 10, 18, and 20 do not exhibit the same patterns as the results for the other datasets – the procedure ReliefF Relevant includes more features compared to its competitors. These three datasets, along with Dataset 12, are datasets where a weak signal is combined with a small number of training instances.

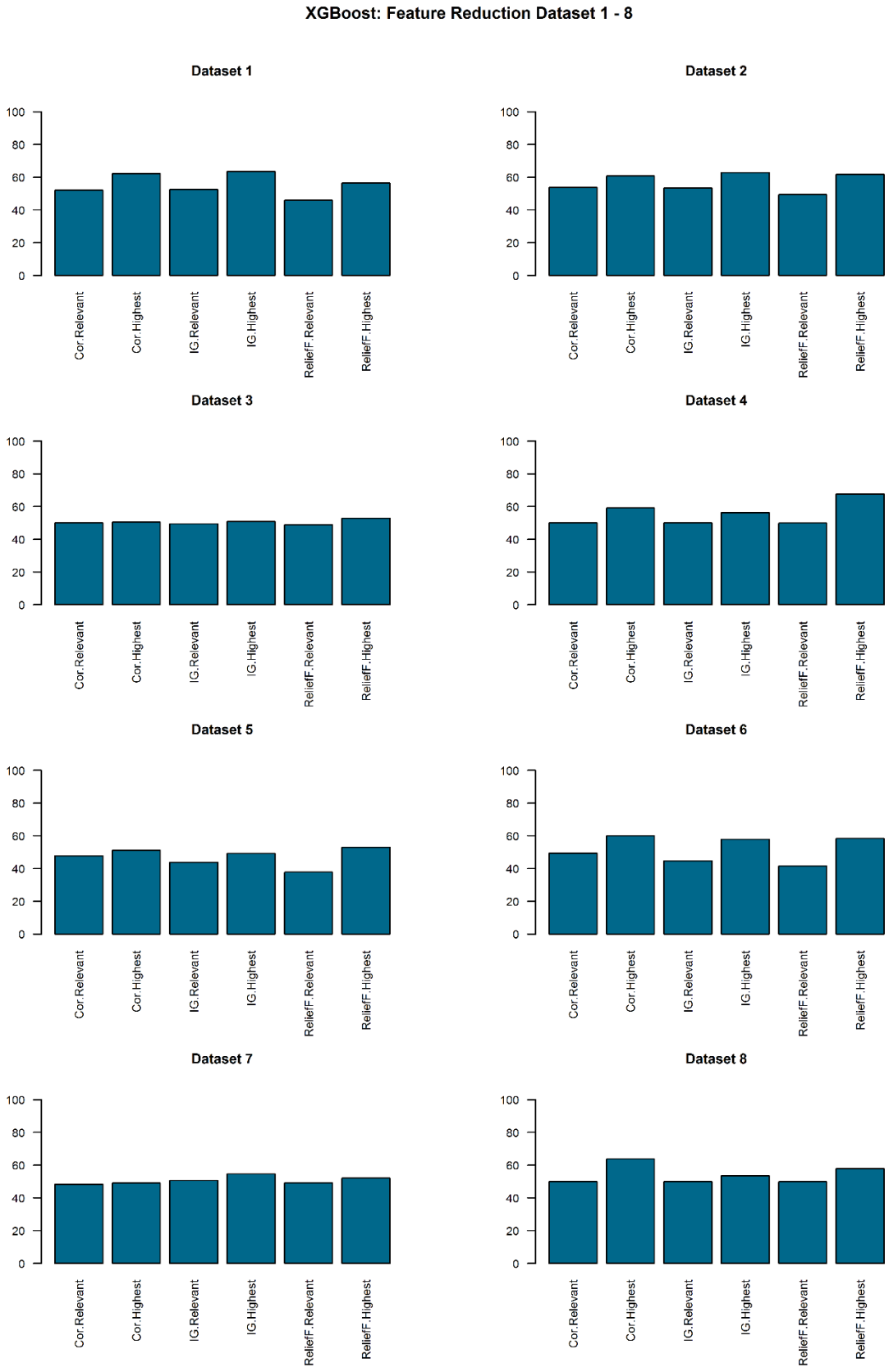


Figure 5.14 Comparison of feature reduction achieved for the XGBoost classifier: Dataset 1 – 8.

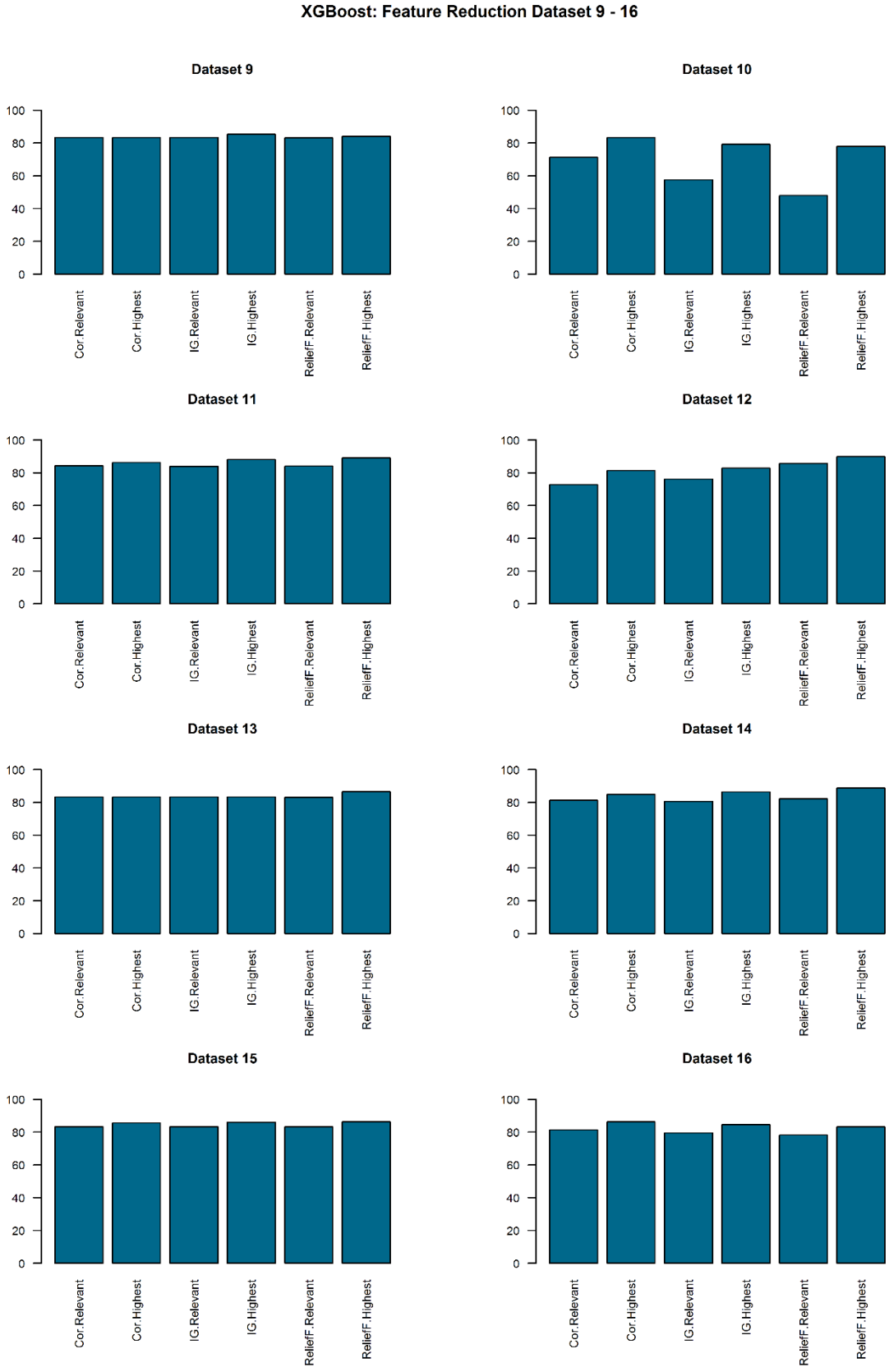


Figure 5.15 Comparison of feature reduction achieved for the XGBoost classifier: Dataset 9 – 16.

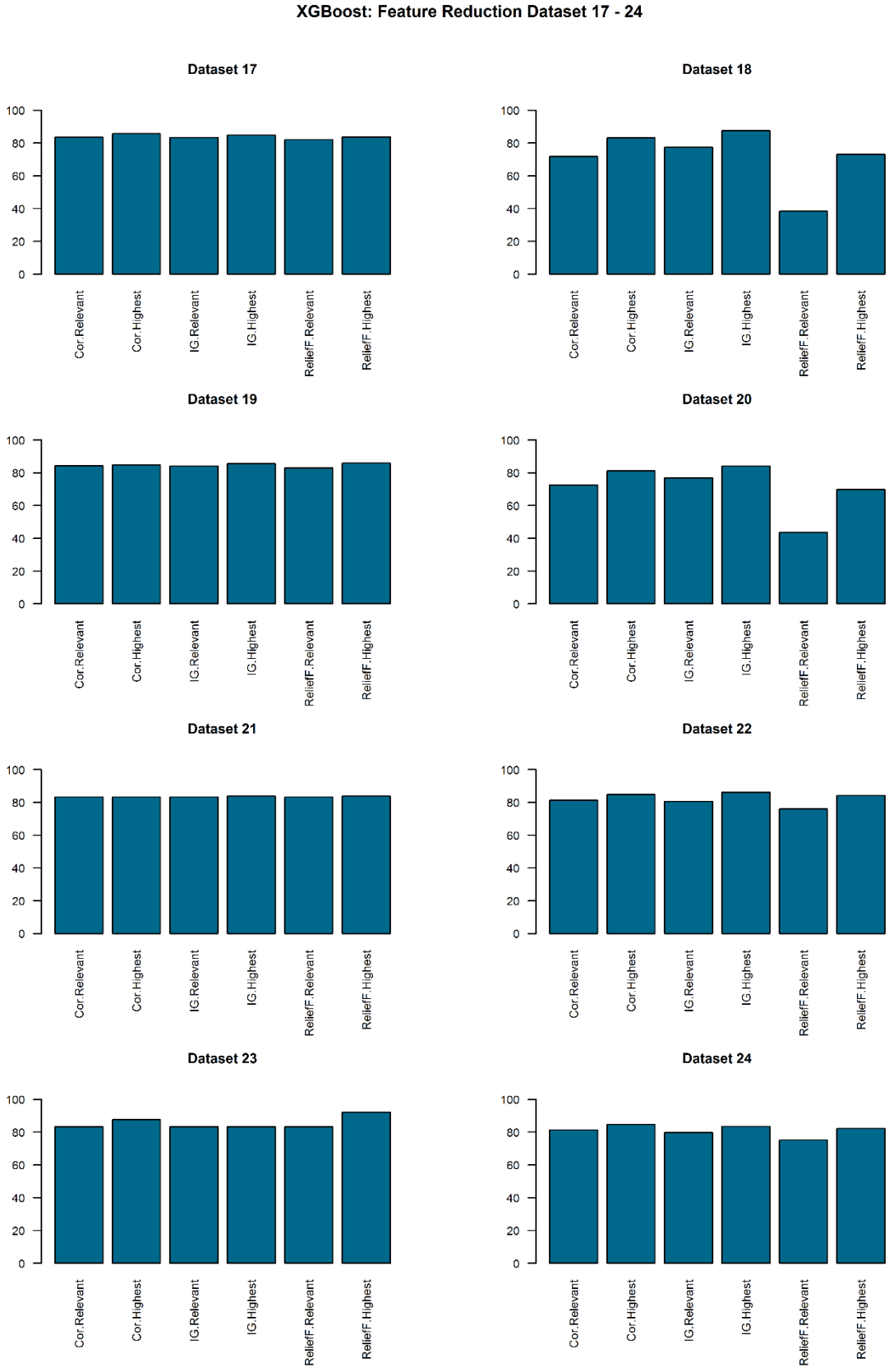


Figure 5.16 Comparison of feature reduction achieved for the XGBoost classifier: Dataset 17 – 24.

Based on the results presented in this section for the XGBoost classifier, some interesting observations are:

1) As was the case for the SVM classifier, the correlation coefficient generally performs better than the two more popular relevance measures, IG and ReliefF when only the relevant features are included.
2) The FS procedure Highest leads to vastly reduced feature sets. The performance of these reduced models compares well to the full model. ReliefF Highest is the only procedure that ranks below the model based on the full set of features, and this is only for one evaluation measure, namely Precision.
3) Unlike the SVM classifier, the XGBoost classifier is not sensitive to smaller values of density. The XGBoost classifier is able to deal with much smaller values for densities, irrespective of the number of training instances used. The classifier was tested successfully on density values less than 0.05.

Based on the results presented in Sections 5.4.1 and 5.4.2, one could ask: *Which classifier performs better when the proposed RPFS procedure is applied?* Or: *Which classifier performs better in the presence of label dependence?* Sections 5.4.3 and 5.4.4 aim to answer questions such as these.

5.4.3 Comparison of the SVM and XGBoost classifiers

In order to compare the SVM and XGBoost classifiers based on the four evaluation measures, namely Hamming-loss, One-error, Precision, and Recall, boxplots will be used. For illustration purposes, boxplots are constructed for each of the FS procedures for Dataset 24 and these are plotted side by side in Figures 5.17 to 5.20. The results for the remaining 23 datasets are available in Appendix F[8].

Following the boxplots, a more formal analysis is conducted using a three-way ANOVA. A summary of the results from the three-way ANOVAs for all 24 datasets is given. The individual ANOVAs per dataset can be found in Appendix G[9].

Turning to the boxplots in Figures 5.17 to 5.20, one can conclude that XGBoost performs better than the SVM in terms of both Hamming-loss and Precision, while the SVM classifier performs

[8] and
[9] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

better for One-error and Recall. For Hamming-loss and One-error, the IQRs for the two procedures are similar, but there is more variation in the results of the XGBoost classifier for Precision and Recall. These conclusions remain the same for all three FS procedures.

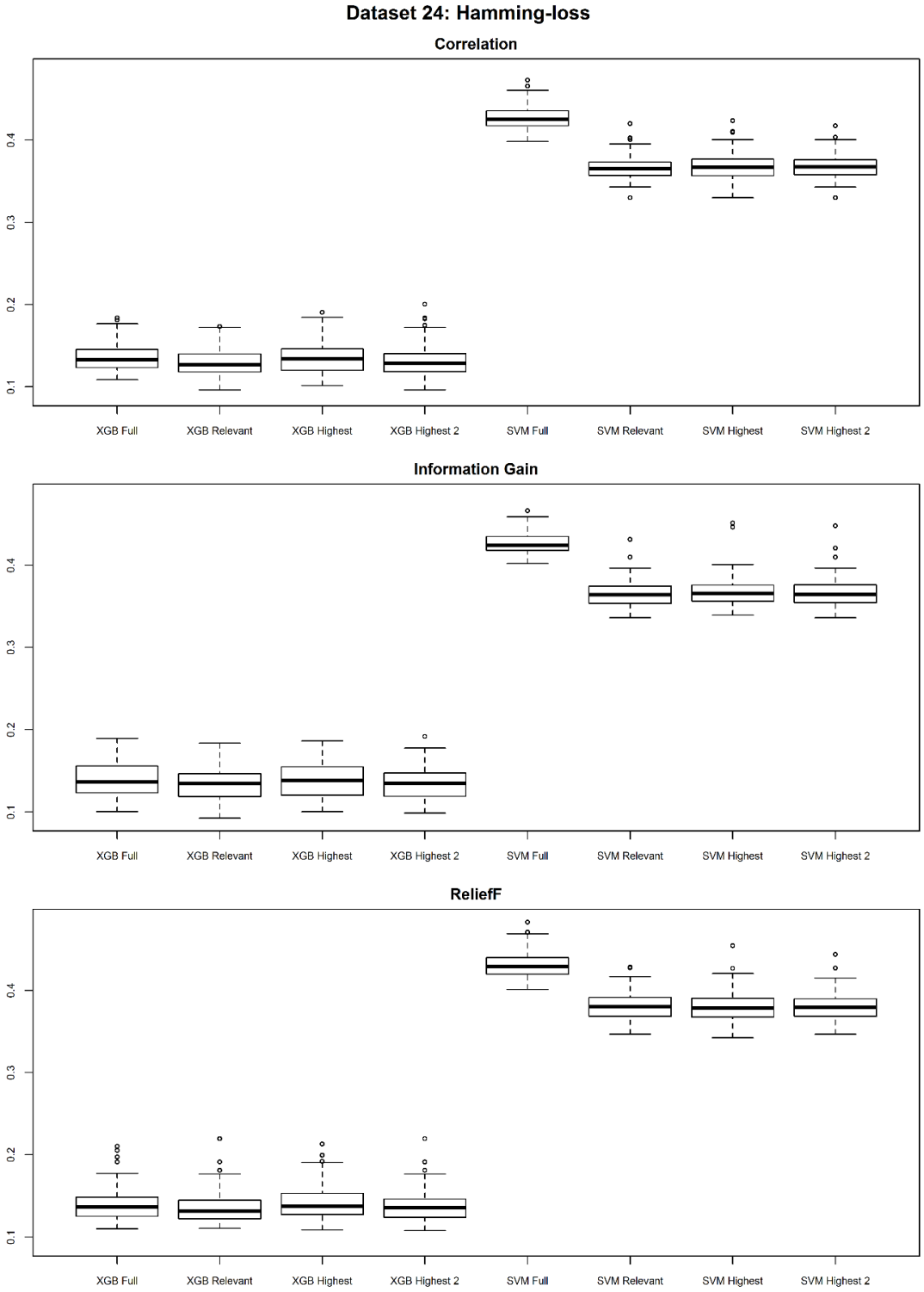


Figure 5.17 Comparing the SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 24.

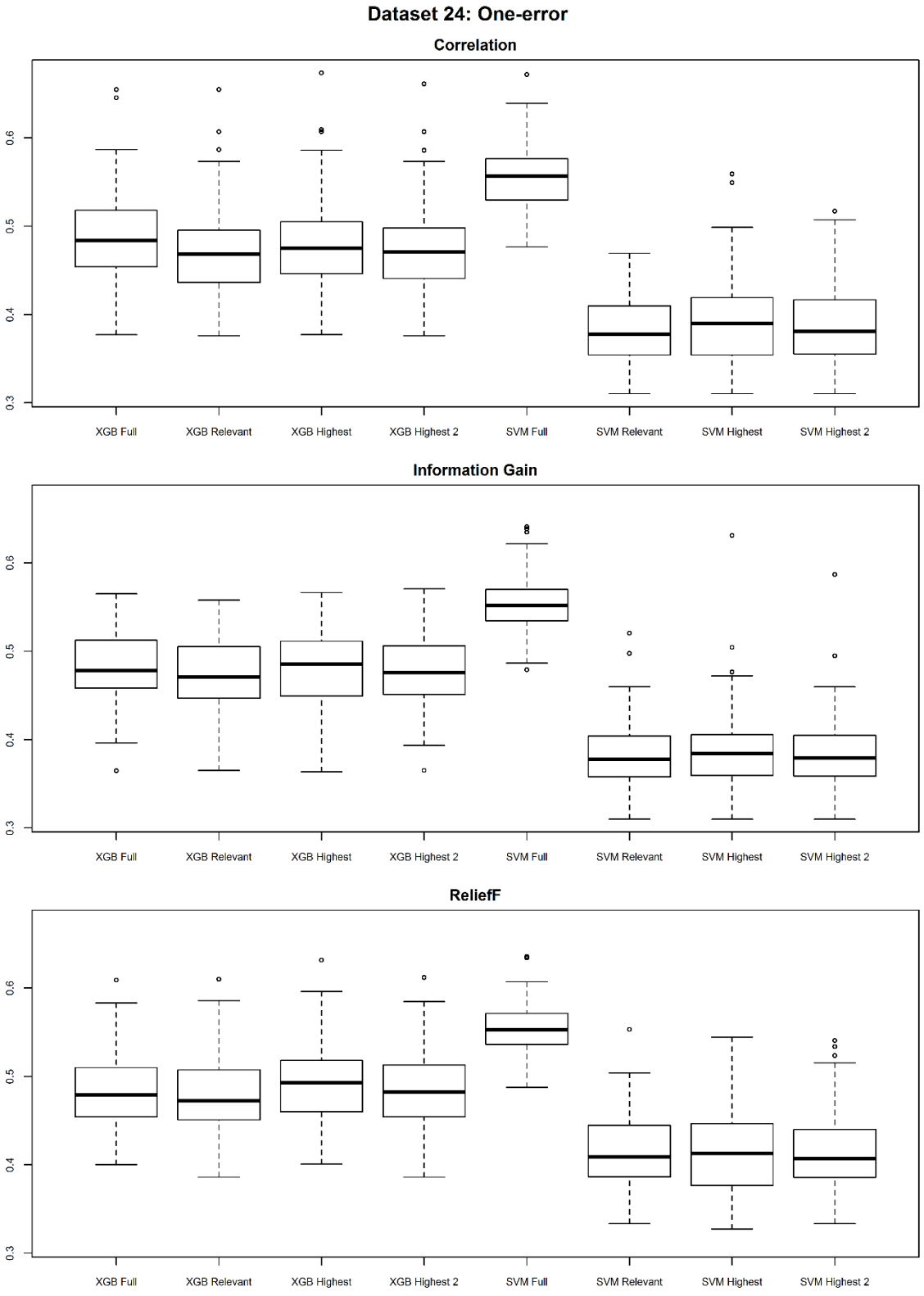


Figure 5.18 Comparing the SVM and XGBoost classifiers with respect to One-error: Dataset 24.

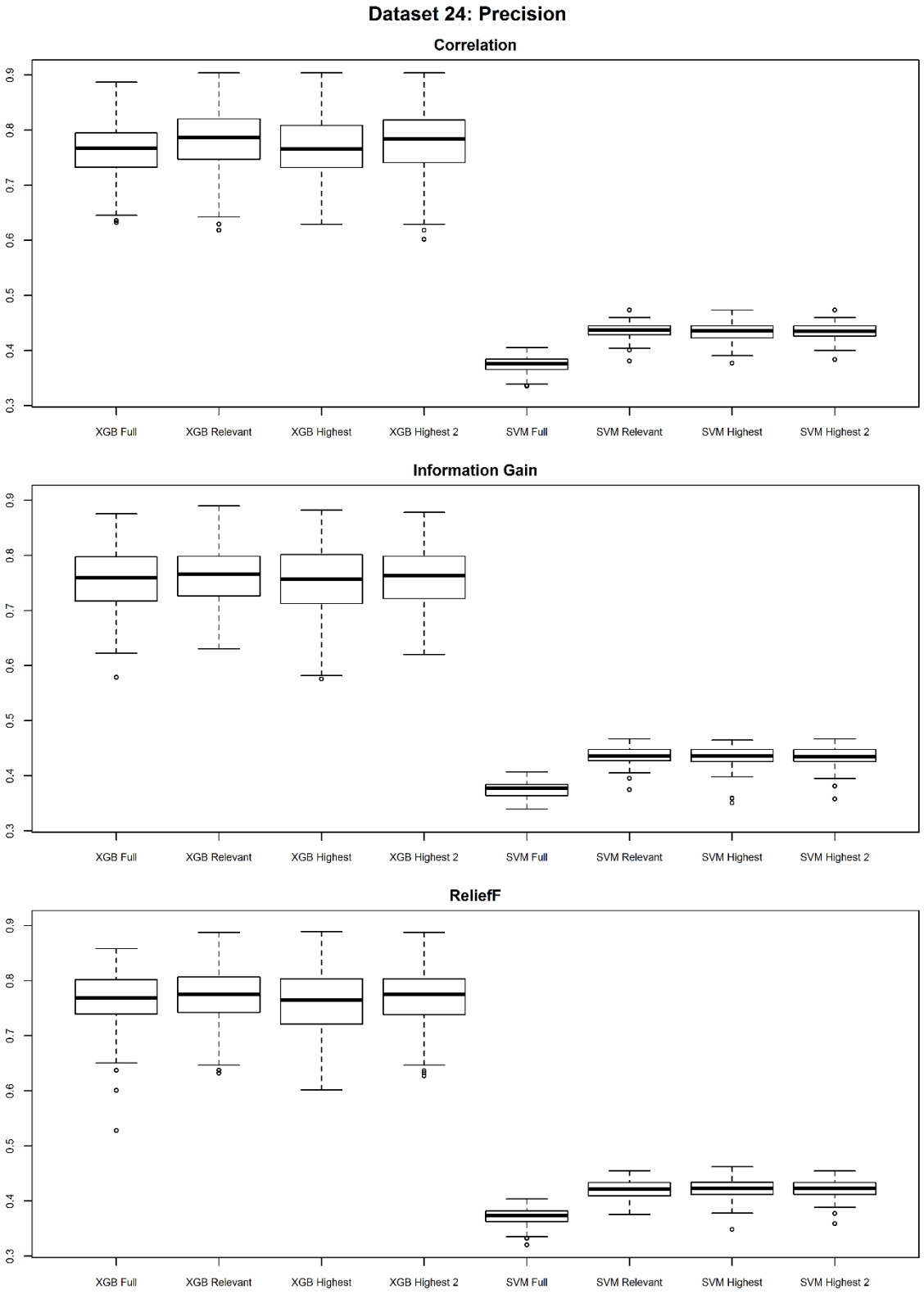


Figure 5.19 Comparing the SVM and XGBoost classifiers with respect to Precision: Dataset 24.

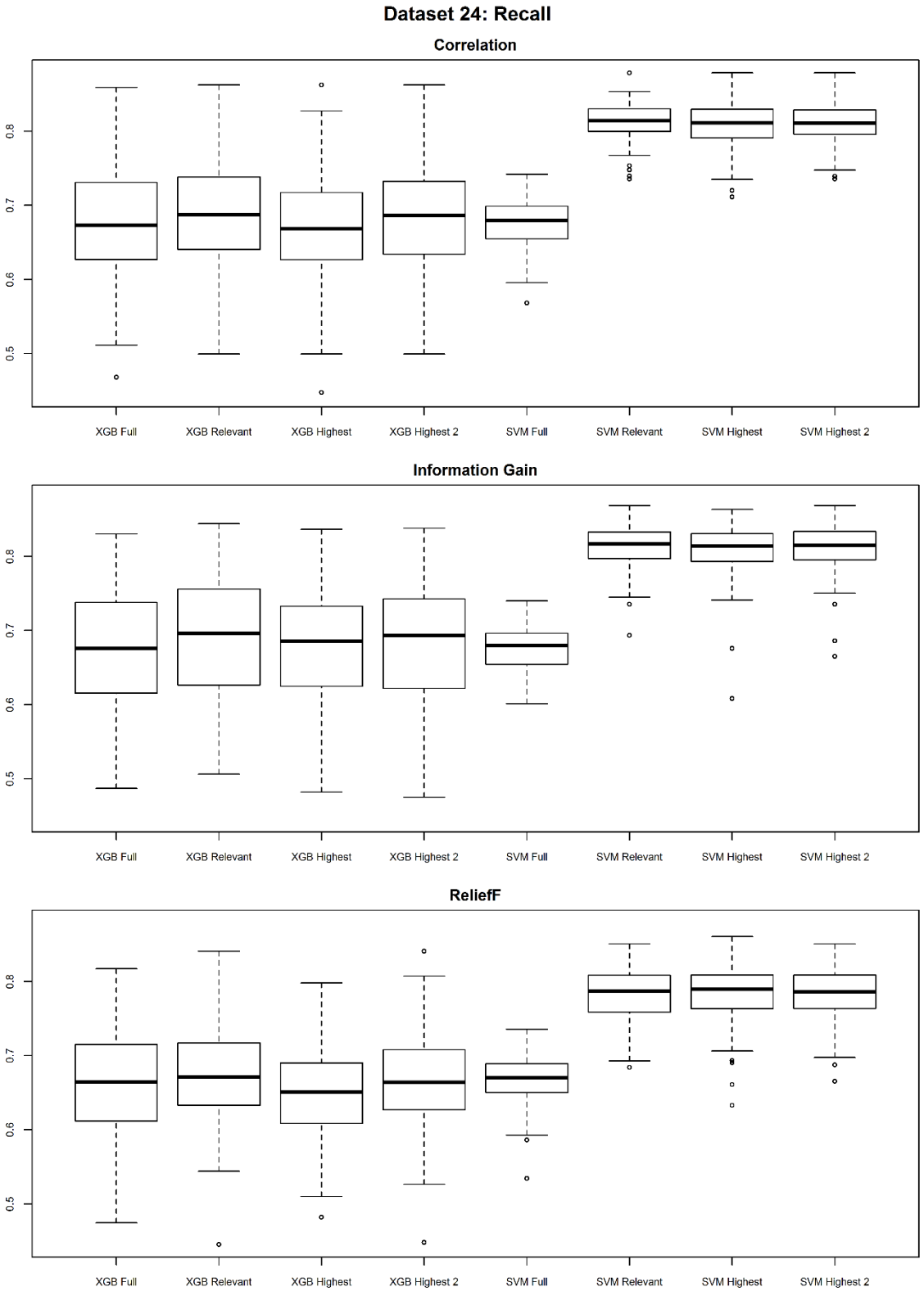


Figure 5.20 Comparing the SVM and XGBoost classifiers with respect to Recall: Dataset 24.

In order to determine whether the differences between the two classifiers observed in Figures 5.17 to 5.20 are statistically significant, a three-way ANOVA is conducted: with factors *Measure*, *Model*, and *Technique*. In Table 5.25, as well as in all future summaries of ANOVA results, *Measure* refers to the relevance measure used, *i.e.* correlation coefficient, IG or ReliefF. Furthermore, *Model* refers to the specific procedure, namely Full, Relevant, Highest, or Highest 2. Finally, *Technique* refers to the classifier used, *i.e.* the SVM or XGBoost. Significant results are shaded pink.

From the three-way ANOVA one is able to conclude that the differences observed between the two classifiers, the three relevance measures, and the four different models are all significant for Dataset 24. The three-way ANOVA also shows that there are significant interactions present. These interactions for One-error are presented in Figures 5.21 and 5.22, but the interpretation of these interactions is difficult and falls outside the scope of this dissertation.

Table 5.25: Three-way ANOVA: Dataset 24.

Hamming-loss

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.029252	0.014626	46.36516	1.77E-20
Model	3	0.401451	0.133817	424.2098	1.1E-220
Technique	1	37.34267	37.34267	118379	0
Measure:Model	6	0.003992	0.000665	2.109171	0.049265
Measure:Technique	2	0.010154	0.005077	16.0944	1.14E-07
Model:Technique	3	0.329104	0.109701	347.7617	3.3E-187
Measure:Model:Technique	6	0.001712	0.000285	0.904292	0.490674
Residuals	2376	0.749509	0.000315		

One-error

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.103064	0.051532	28.47363	6.02E-13
Model	3	3.011235	1.003745	554.611	3.5E-273
Technique	1	1.257398	1.257398	694.7648	1.6E-134
Measure:Model	6	0.047106	0.007851	4.337977	0.000232
Measure:Technique	2	0.044895	0.022448	12.40321	4.38E-06
Model:Technique	3	2.628356	0.876119	484.0921	1.9E-245
Measure:Model:Technique	6	0.016414	0.002736	1.511592	0.170237
Residuals	2376	4.300128	0.00181		

Precision

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.037191	0.018596	10.04129	4.54E-05
Model	3	0.431163	0.143721	77.60674	6.87E-48
Technique	1	73.14939	73.14939	39499.39	0
Measure:Model	6	0.005587	0.000931	0.502825	0.806638
Measure:Technique	2	0.040493	0.020247	10.93285	1.88E-05
Model:Technique	3	0.316111	0.10537	56.89815	1.62E-35
Measure:Model:Technique	6	0.003215	0.000536	0.289376	0.942223
Residuals	2376	4.400143	0.001852		

Recall

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.206496	0.103248	31.78795	2.38E-14
Model	3	2.068368	0.689456	212.2695	5.5E-122
Technique	1	5.313725	5.313725	1635.988	1.3E-272
Measure:Model	6	0.018757	0.003126	0.962494	0.449115
Measure:Technique	2	0.006462	0.003231	0.994807	0.369949
Model:Technique	3	1.774922	0.591641	182.1541	2.6E-106
Measure:Model:Technique	6	0.007812	0.001302	0.400853	0.878844
Residuals	2376	7.717301	0.003248		

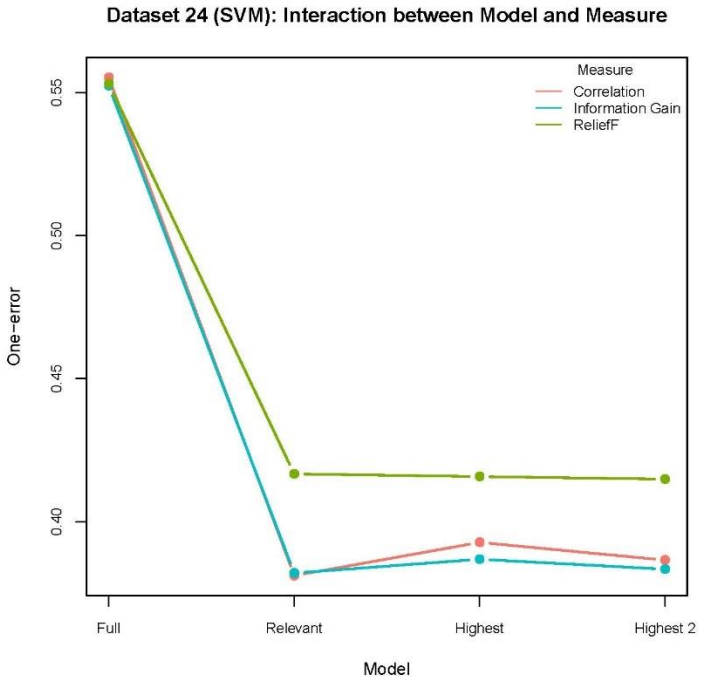


Figure 5.21 Interaction between *Model* and *Measure* for SVM classifier: Dataset 24.

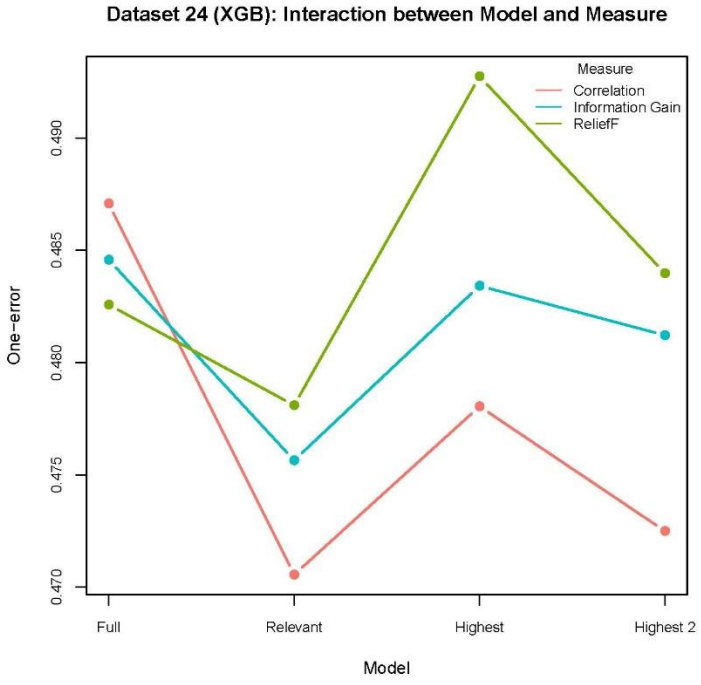


Figure 5.22 Interaction between *Model* and *Measure* for XGBoost classifier: Dataset 24.

Table 5.26: Summary of the p -values of the main effects.

	Hamming-loss			Recall			Precision			One-Error		
	Measure	Model	Technique	Measure	Model	Technique	Measure	Model	Technique	Measure	Model	Technique
Dataset 1	0.001	0.000	0.000	0.021	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 2	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000
Dataset 3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 5	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
Dataset 6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 11	0.000	0.000	0.000	0.000	0.000	0.000	0.178	0.000	0.000	0.000	0.000	0.000
Dataset 12	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 14	0.000	0.000	0.000	0.081	0.000	0.000	0.017	0.000	0.000	0.014	0.000	0.000
Dataset 15	0.095	0.000	0.000	0.441	0.000	0.000	0.062	0.000	0.000	0.000	0.000	0.000
Dataset 16	0.066	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 19	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 20	0.000	0.000	0.000	0.421	0.000	0.000	0.161	0.000	0.000	0.152	0.000	0.000
Dataset 21	0.189	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dataset 24	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Since only the main effects are of interest, a summary of the main effects of the three-way ANOVAs for all 24 datasets is provided in Table 5.26. The table includes the p-values for each of the ANOVAs. Differences that are not significant are shaded green. One can therefore conclude that the majority of differences are significant for all of the synthetic datasets.

For Dataset 24, XGBoost performs better than the SVM for both Hamming-loss and Precision, while the SVM classifier performs better for One-error and Recall. The boxplots for the other 23 datasets are presented in Appendix F[10]. The majority of these datasets present the same findings as in the case of Dataset 24. The few exceptions are described briefly below.

When one considers One-error and Recall for Dataset 7, the XGBoost classifier performs better than the SVM classifier when IG is used as relevance measure. For Dataset 15, the XGBoost classifier performs better than the SVM classifier for all three relevance measures for Hamming-loss, Precision, and Recall. A similar situation occurs when one considers Recall for Dataset 23: some of the procedures implementing XGBoost perform better than the procedures based on the SVM classifier.

In the next section the effect on the performances of the two classifiers of the properties defining the 24 datasets is investigated.

5.4.4 Evaluating the performance of the proposed feature selection procedures

In this section the SVM and XGBoost are compared taking into account the four characteristics that are used to define the different datasets, namely the signal-to-noise ratio, label dependence, number of irrelevant features, and different density vectors. Datasets 1 and 3 are compared to investigate the influence of the signal strength on the relative performance of the two classifiers. Similarly, Datasets 1 and 9 are compared to study the influence of the number of irrelevant features on the relative performance of the SVM and XGBoost. The effect of the presence of label dependence is studied by comparing the results obtained for Datasets 10 and 12. Finally, Datasets 9 and 17 are compared to determine whether the performances are influenced by the vectors of densities. A four-way ANOVA is conducted to complete this section. For each subsection, the structure of the two datasets is described and then boxplots are presented to compare the two classifiers.

[10] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

In the next subsection, the relative performances of the SVM classifier and XGBoost are compared based on different signal-to-noise ratios.

Comparing performance of techniques at different signal-to-noise ratios

Datasets 1 and 3 are identical except for the signal strength (see Table 5.27). Dataset 3 has a larger signal-to-noise ratio than Dataset 1 and comparing the results obtained for these two datasets therefore allows one to draw conclusions regarding the effect of a change in the signal-to-noise ratio. On general grounds one would expect the performance of the classifiers to improve as the signal becomes stronger.

Table 5.27: Structure of Dataset 1 and Dataset 3.

	k	$p - k$	q	ρ	**Signal**	**Vector of Densities**	**Number of training instances**	**Number of test instances**
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 3	10	10	6	0	100	0.4	80	10 000

Based on Figures 5.23 to 5.26, one can conclude that both the classifiers perform better when the signal-to-noise ratio increases from 10 to 100. This is the case across all three relevance measures and all four evaluation measures. It is interesting to note that the variation in the results using the SVM classifier is smaller for Dataset 3 than for Dataset 1.

In Table 5.29, as well as all further summaries of four-way ANOVA results, *Measure* refers to the relevance measure used, *i.e.* correlation coefficient, IG, or ReliefF. Furthermore, *Model* refers to the features included in the procedure, namely Full, Relevant, Highest, or Highest 2, whilst *Technique* refers to the classifier used, *i.e.* the SVM or XGBoost. Finally, *Dataset* refers to the two datasets that are being compared, for example Dataset 1 and Dataset 3. All significant p-values in the four-way ANOVA are shaded pink.

The summary in Table 5.29 of the results of the four-way ANOVA allows one to conclude that the main effects corresponding to the differences observed between the two classifiers, the three relevance measures, the four different models, and the two datasets are all significant.

One can also conclude that the performance of the procedures does improve as the signal becomes stronger.

If one considers another set of datasets that differ only with regard to the signal strength, such as Dataset 17 and Dataset 21, one would expect to see similar results. The structure of these two datasets is provided in Table 5.28.

Table 5.28: Structure of Dataset 17 and Dataset 21.

	k	$p - k$	q	ρ	**Signal**	**Vector of Densities**	**Number of training instances**	**Number of test instances**
Dataset 17	50	10	6	0	10	vary	240	10 000
Dataset 21	50	10	6	0	100	vary	240	10 000

The associated boxplots and the results from the four-way ANOVA can be found in Appendices H and I[11], respectively. From these results it once again seems that the performances of the procedures improve as the signal-to-noise ratio increases, as expected.

[11] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

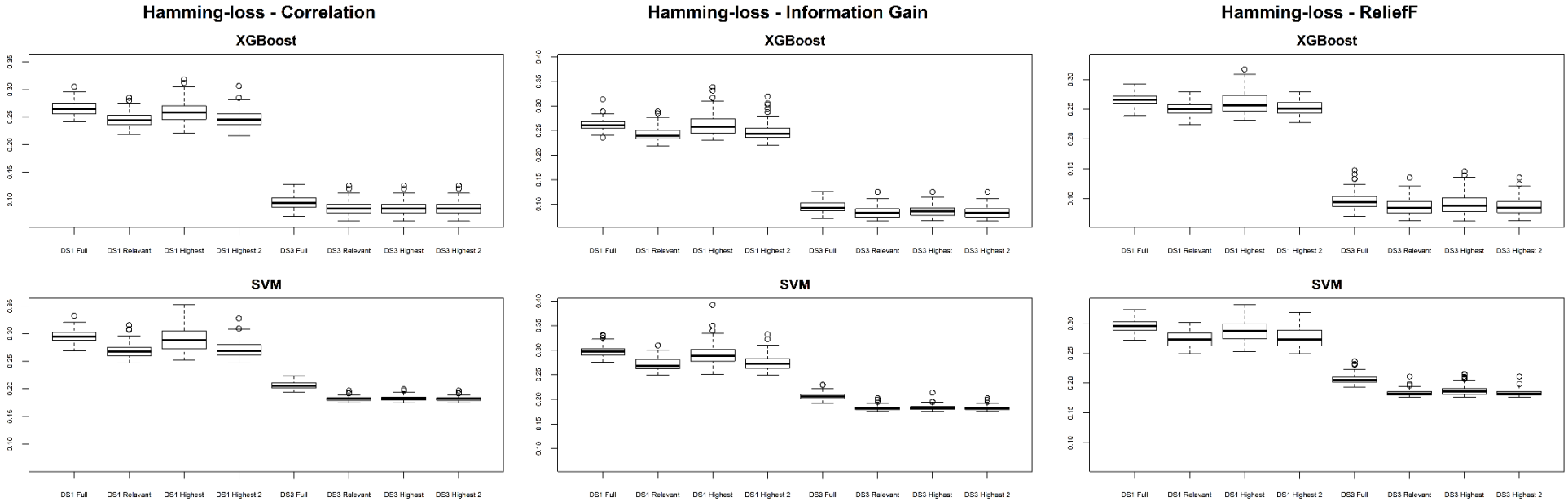


Figure 5.23 Hamming-loss: Dataset 1 vs Dataset 3.

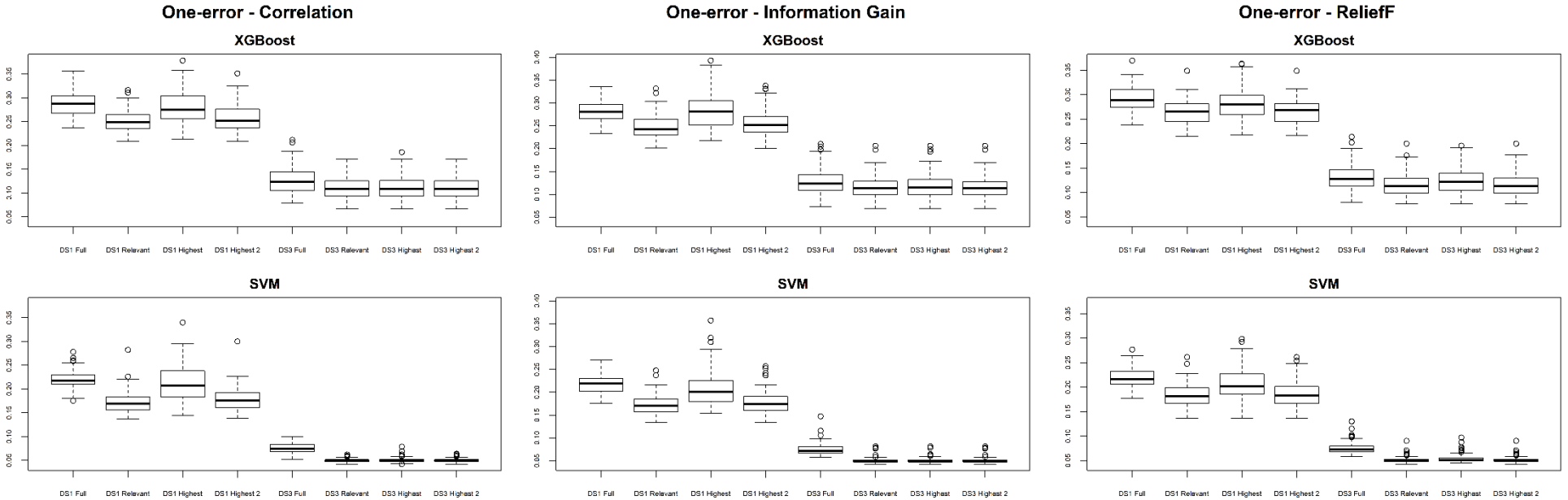


Figure 5.24 One-error: Dataset 1 vs Dataset 3.

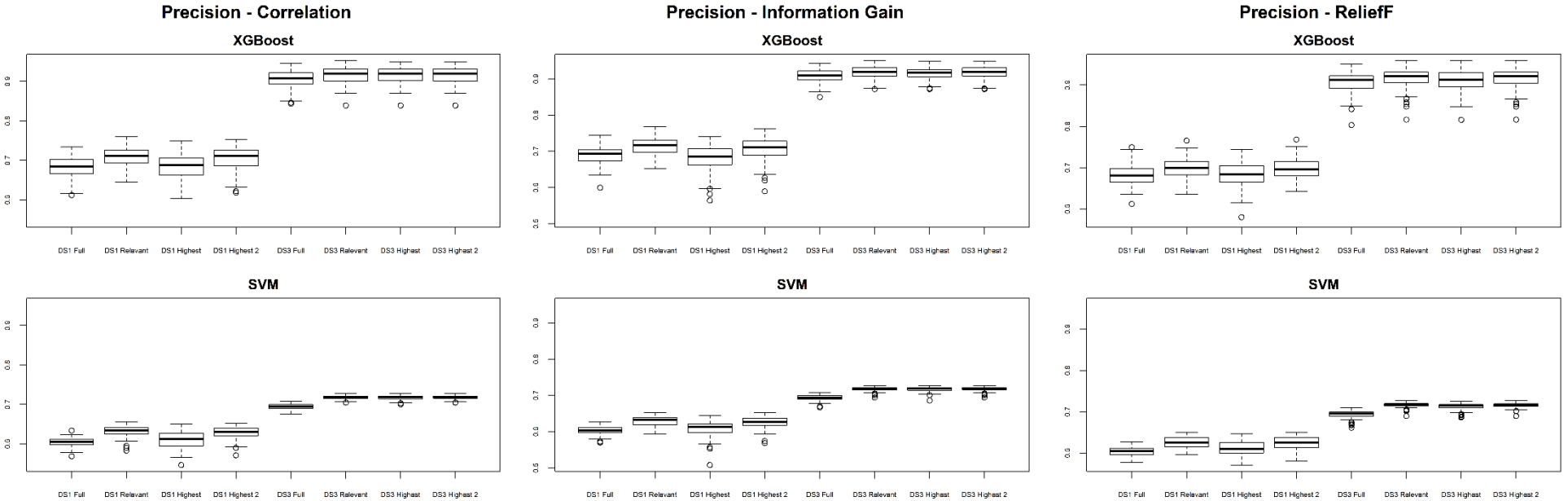


Figure 5.25 Precision: Dataset 1 vs Dataset 3.

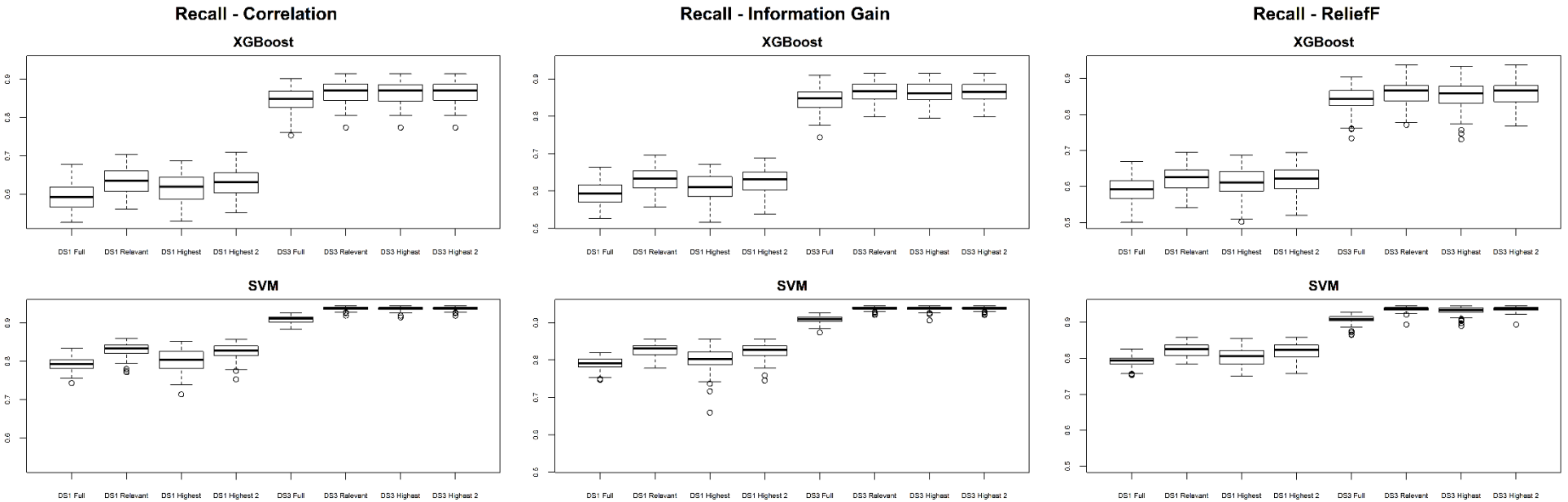


Figure 5.26 Recall: Dataset 1 vs Dataset 3.

Table 5.29: Four-way ANOVA: Dataset 1 vs Dataset 3.

Hamming-loss

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006091	0.003045	18.11888	1.45E-08
Model	3	0.290548	0.096849	576.2337	0
Technique	1	4.943154	4.943154	29410.75	0
Dataset	1	20.38022	20.38022	121258.1	0
Measure:Model	6	0.001577	0.000263	1.563453	0.15354
Measure:Technique	2	0.000663	0.000332	1.972461	0.139228
Model:Technique	3	0.020936	0.006979	41.522	1.77E-26
Measure:Dataset	2	7.62E-05	3.81E-05	0.226654	0.797205
Model:Dataset	3	0.035177	0.011726	69.7644	3.8E-44
Technique:Dataset	1	1.594658	1.594658	9487.888	0
Measure:Model:Technique	6	0.000342	5.69E-05	0.338824	0.916604
Measure:Model:Dataset	6	0.004849	0.000808	4.808292	6.71E-05
Measure:Technique:Dataset	2	0.000182	9.11E-05	0.542028	0.581604
Model:Technique:Dataset	3	0.004036	0.001345	8.005134	2.55E-05
Measure:Model:Technique:Dataset	6	0.000144	2.41E-05	0.143179	0.990387
Residuals	4752	0.798683	0.000168		

One-error

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.018974	0.009487	17.14929	3.79E-08
Model	3	0.6325	0.210833	381.1154	8.2E-222
Technique	1	5.649046	5.649046	10211.56	0
Dataset	1	25.28774	25.28774	45711.69	0
Measure:Model	6	0.006572	0.001095	1.979961	0.064934
Measure:Technique	2	0.002361	0.00118	2.133701	0.118512
Model:Technique	3	0.026484	0.008828	15.95786	2.54E-10
Measure:Dataset	2	0.005331	0.002666	4.818415	0.008119
Model:Dataset	3	0.144892	0.048297	87.30539	5.48E-55
Technique:Dataset	1	0.047654	0.047654	86.14315	2.48E-20
Measure:Model:Technique	6	0.001195	0.000199	0.360116	0.904301
Measure:Model:Dataset	6	0.012032	0.002005	3.62491	0.001367
Measure:Technique:Dataset	2	0.001666	0.000833	1.505455	0.222022
Model:Technique:Dataset	3	0.003601	0.0012	2.17007	0.089409
Measure:Model:Technique:Dataset	6	0.000599	9.98E-05	0.18049	0.982258
Residuals	4752	2.62881	0.000553		

Precision

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010536	0.005268	13.75763	1.1E-06
Model	3	0.324615	0.108205	282.5758	8.8E-169
Technique	1	23.42165	23.42165	61165.35	0
Dataset	1	29.32038	29.32038	76569.8	0
Measure:Model	6	0.002681	0.000447	1.166833	0.320959
Measure:Technique	2	0.002735	0.001367	3.570676	0.028212
Model:Technique	3	0.024125	0.008042	21.00051	1.64E-13
Measure:Dataset	2	0.000536	0.000268	0.700029	0.496622
Model:Dataset	3	0.061571	0.020524	53.59748	4.6E-34
Technique:Dataset	1	4.682518	4.682518	12228.34	0
Measure:Model:Technique	6	0.000942	0.000157	0.409976	0.872889
Measure:Model:Dataset	6	0.007559	0.00126	3.28995	0.003122
Measure:Technique:Dataset	2	0.00176	0.00088	2.298511	0.10052
Model:Technique:Dataset	3	0.004288	0.001429	3.732355	0.010766
Measure:Model:Technique:Dataset	6	0.000709	0.000118	0.308722	0.932731
Residuals	4752	1.819653	0.000383		

Recall

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.01797	0.008985	14.18336	7.22E-07
Model	3	0.683852	0.227951	359.8359	1.4E-210
Technique	1	21.5256	21.5256	33979.63	0
Dataset	1	39.38182	39.38182	62166.9	0
Measure:Model	6	0.003513	0.000586	0.924253	0.476081
Measure:Technique	2	0.003331	0.001666	2.629373	0.072229
Model:Technique	3	0.004001	0.001334	2.105528	0.097334
Measure:Dataset	2	0.001332	0.000666	1.051629	0.34945
Model:Dataset	3	0.05921	0.019737	31.15563	6.15E-20
Technique:Dataset	1	4.595125	4.595125	7253.719	0
Measure:Model:Technique	6	0.000593	9.88E-05	0.15602	0.987908
Measure:Model:Dataset	6	0.008765	0.001461	2.306085	0.031686
Measure:Technique:Dataset	2	0.00036	0.00018	0.284042	0.752748
Model:Technique:Dataset	3	0.015292	0.005097	8.046648	2.4E-05
Measure:Model:Technique:Dataset	6	0.000266	4.44E-05	0.070031	0.998677
Residuals	4752	3.010323	0.000633		

In the next subsection the relative performances of the SVM classifier and XGBoost are reported based on the number of irrelevant features in the dataset.

Comparing performance of techniques with respect to number of irrelevant features

In order to investigate the influence of the number of irrelevant features on the performance of the two classification procedures, Datasets 1 and 9 are compared (refer to Table 5.30). Dataset 1 only has ten irrelevant features, while Dataset 9 has 50. It is important to remember that the number of training instances are dependent on the number of features, *i.e.* $4p$. This implies that the datasets also differ with regard to the number of training instances. Of interest is to determine whether the presence of irrelevant features negatively impacts the performance of the SVM and XGBoost.

Table 5.30: Structure of Dataset 1 and Dataset 9.

	k	$p-k$	q	ρ	**Signal**	**Vector of Densities**	**Number of training instances**	**Number of test instances**
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 9	50	10	6	0	10	0.4	240	10 000

From Figures 5.27 to 5.30 it can be seen that the performances of the procedures are not influenced negatively by the presence of a higher proportion of irrelevant features. It is interesting that the variation in results decreases as the number of irrelevant features increases. This result is counterintuitive as one would expect that a larger number of irrelevant features would have a negative impact on performance. One possible explanation could be the larger number of training instances associated with Dataset 9. It would be reasonable to expect that the performances would improve if the number of training instances increases.

Table 5.32 shows the results from the four-way ANOVA. From this table, one can conclude that the differences observed between the two classifiers, the three relevance measures, the four different models, and the two datasets are all significant. It is also confirmed that the performances of the two classifiers improve as the number of training instances increases, even if the proportion of irrelevant features is increased.

To confirm these results, a second pair of datasets are compared. Datasets 5 and 17 have the same number of features as Datasets 1 and 9, but with different vectors of densities. The configuration is given in Table 5.31.

Table 5.31: Structure of Dataset 5 and Dataset 17.

	k	$p - k$	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 5	10	10	6	0	10	vary	80	10 000
Dataset 17	50	10	6	0	10	vary	240	10 000

The associated boxplots and results from the four-way ANOVA comparing these two datasets can be found in Appendices H and I, respectively. These results confirm that the performances of the procedures improve as the number of training instances increases in spite of the presence of a larger proportion of irrelevant features.

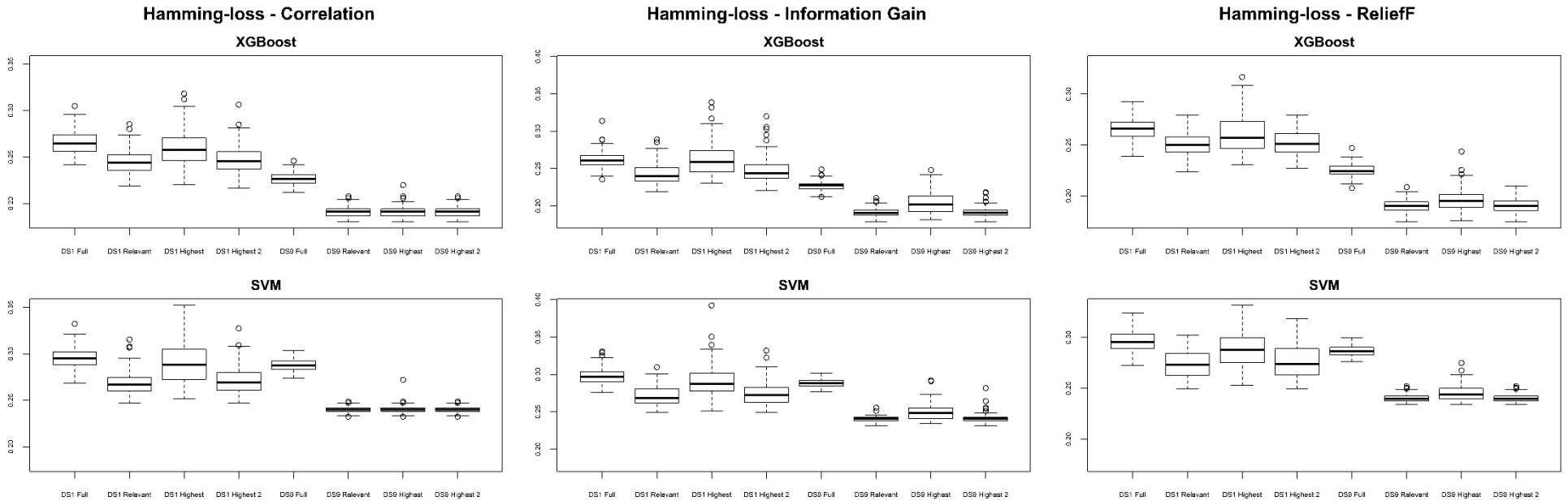


Figure 5.27 Hamming-loss: Dataset 1 vs Dataset 9.

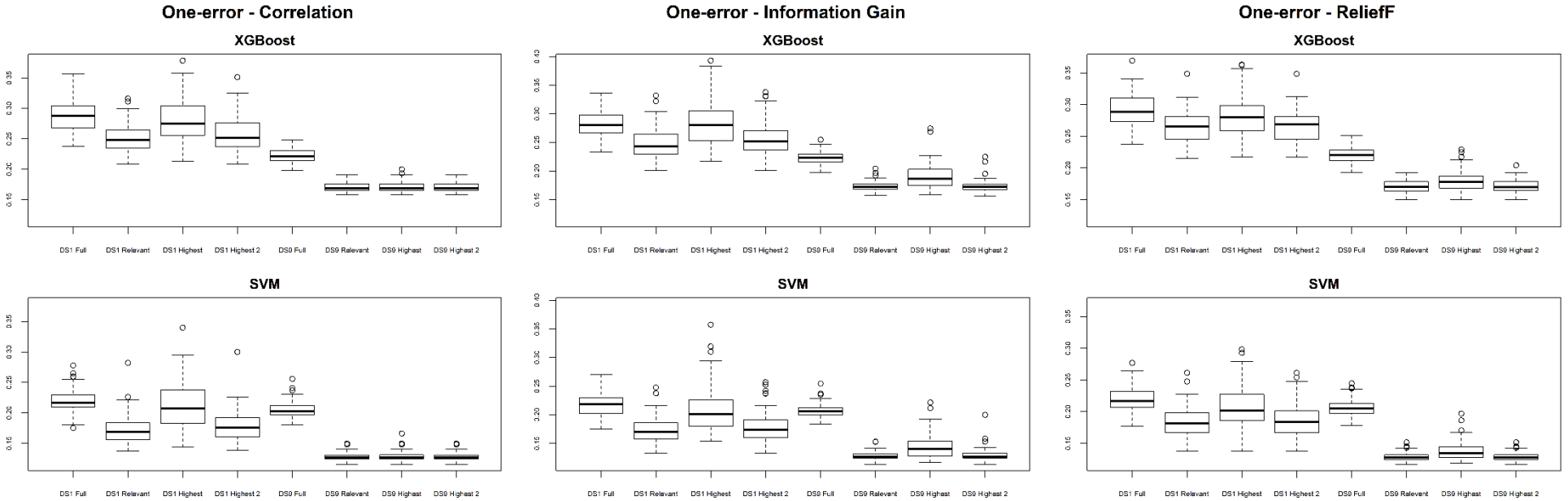


Figure 5.28 One-error: Dataset 1 vs Dataset 9.

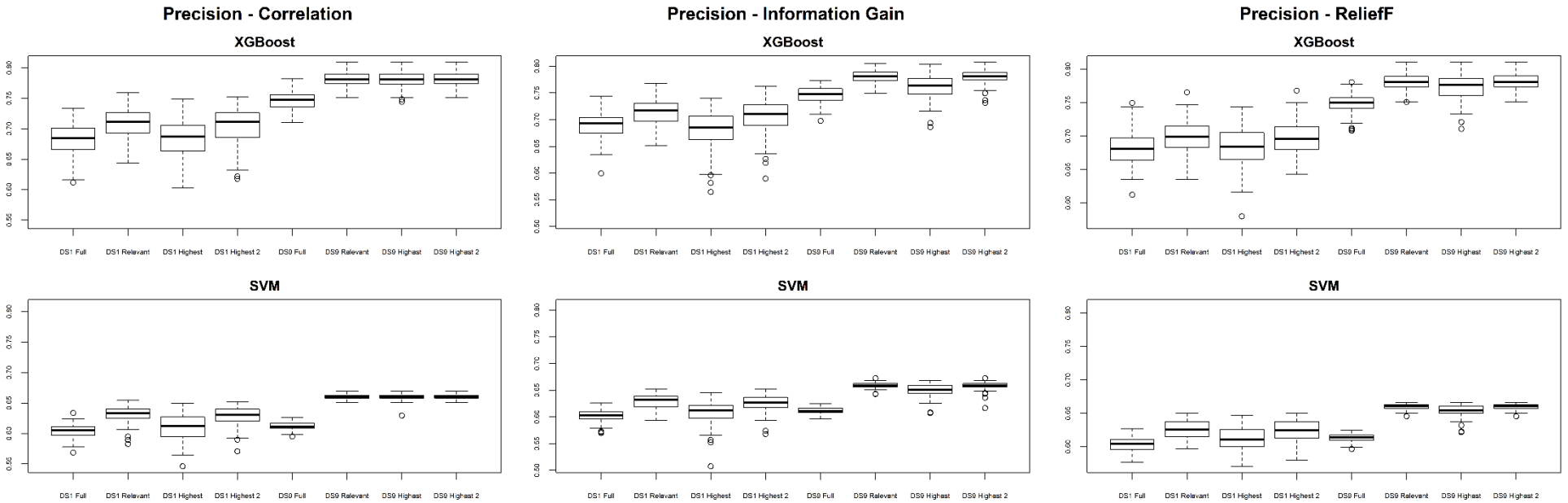


Figure 5.29 Precision: Dataset 1 vs Dataset 9.

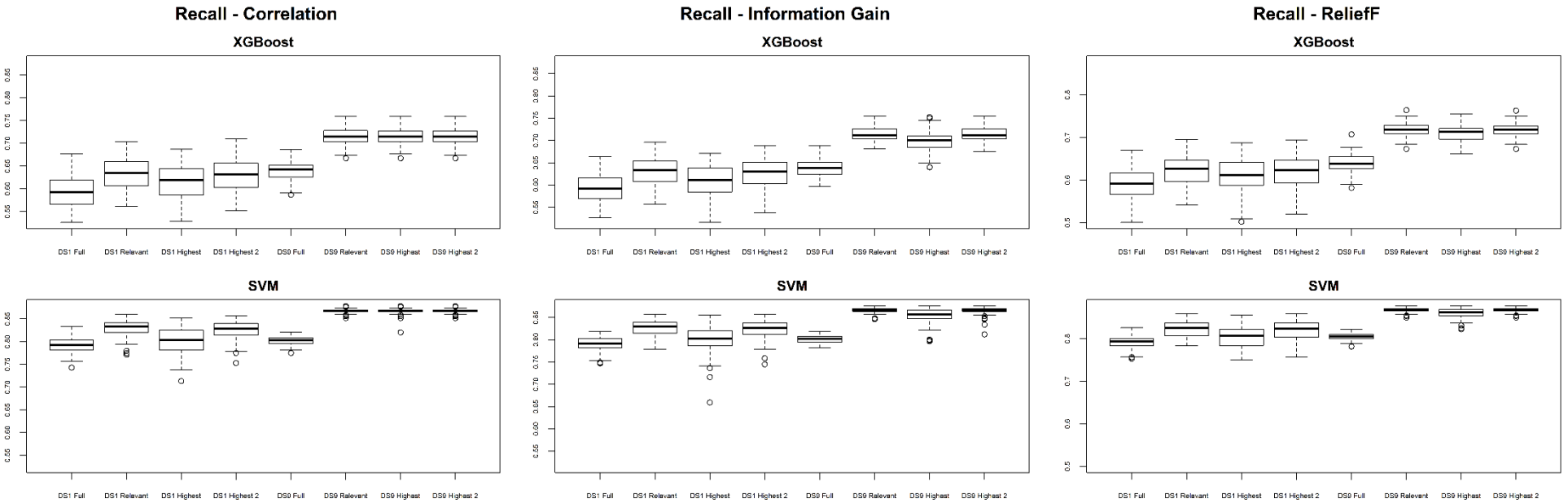


Figure 5.30 Recall: Dataset 1 vs Dataset 9.

Table 5.32: Four-way ANOVA: Dataset 1 vs Dataset 9.

Hamming-loss

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004043	0.002022	14.64135	4.58E-07
Model	3	0.763221	0.254407	1842.533	0
Technique	1	1.918268	1.918268	13892.97	0
Dataset	1	2.084919	2.084919	15099.94	0
Measure:Model	6	0.00779	0.001298	9.403007	2.77E-10
Measure:Technique	2	0.000146	7.32E-05	0.530067	0.588601
Model:Technique	3	0.019327	0.006442	46.65712	1.04E-29
Measure:Dataset	2	0.003866	0.001933	13.99957	8.67E-07
Model:Dataset	3	0.14868	0.04956	358.9353	4.1E-210
Technique:Dataset	1	0.180194	0.180194	1305.048	1E-252
Measure:Model:Technique	6	0.000519	8.64E-05	0.62592	0.709707
Measure:Model:Dataset	6	0.005614	0.000936	6.776995	3.62E-07
Measure:Technique:Dataset	2	0.000711	0.000356	2.57543	0.076227
Model:Technique:Dataset	3	0.003511	0.00117	8.476724	1.3E-05
Measure:Model:Technique:Dataset	6	0.000129	2.14E-05	0.155118	0.988093
Residuals	4752	0.656131	0.000138		

One-error

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010621	0.00531	12.43001	4.13E-06
Model	3	1.985963	0.661988	1549.511	0
Technique	1	3.73064	3.73064	8732.292	0
Dataset	1	5.110752	5.110752	11962.72	0
Measure:Model	6	0.020273	0.003379	7.908719	1.68E-08
Measure:Technique	2	2.05E-05	1.02E-05	0.023962	0.976323
Model:Technique	3	0.078658	0.026219	61.37149	6.36E-39
Measure:Dataset	2	0.018182	0.009091	21.27911	6.31E-10
Model:Dataset	3	0.352089	0.117363	274.7113	2E-164
Technique:Dataset	1	0.440345	0.440345	1030.713	7.4E-205
Measure:Model:Technique	6	0.001611	0.000269	0.628668	0.707486
Measure:Model:Dataset	6	0.01686	0.00281	6.577228	6.2E-07
Measure:Technique:Dataset	2	0.001488	0.000744	1.741833	0.175311
Model:Technique:Dataset	3	0.023179	0.007726	18.08502	1.15E-11
Measure:Model:Technique:Dataset	6	0.000619	0.000103	0.241514	0.96278
Residuals	4752	2.030166	0.000427		

Precision

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.005467	0.002734	8.529125	0.000201
Model	3	0.786232	0.262077	817.7165	0
Technique	1	12.10614	12.10614	37772.78	0
Dataset	1	3.343844	3.343844	10433.24	0
Measure:Model	6	0.014767	0.002461	7.679187	3.14E-08
Measure:Technique	2	0.00127	0.000635	1.980761	0.138078
Model:Technique	3	0.029649	0.009883	30.83589	9.79E-20
Measure:Dataset	2	0.009343	0.004671	14.57544	4.89E-07
Model:Dataset	3	0.146041	0.04868	151.889	5.33E-94
Technique:Dataset	1	0.645948	0.645948	2015.442	0
Measure:Model:Technique	6	0.002919	0.000486	1.517839	0.167914
Measure:Model:Dataset	6	0.008379	0.001396	4.357247	0.000215
Measure:Technique:Dataset	2	0.003391	0.001696	5.290808	0.005067
Model:Technique:Dataset	3	0.005216	0.001739	5.42462	0.001009
Measure:Model:Technique:Dataset	6	0.000323	5.39E-05	0.168161	0.985258
Residuals	4752	1.523012	0.00032		

Recall

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.011318	0.005659	11.38104	1.17E-05
Model	3	2.196604	0.732201	1472.606	0
Technique	1	36.79046	36.79046	73993.08	0
Dataset	1	4.224117	4.224117	8495.555	0
Measure:Model	6	0.013507	0.002251	4.52769	0.000139
Measure:Technique	2	0.000322	0.000161	0.323711	0.723475
Model:Technique	3	0.015811	0.00527	10.59944	6.07E-07
Measure:Dataset	2	0.00676	0.00338	6.798138	0.001127
Model:Dataset	3	0.388926	0.129642	260.7361	1.2E-156
Technique:Dataset	1	0.515063	0.515063	1035.896	8.8E-206
Measure:Model:Technique	6	0.000893	0.000149	0.299324	0.937442
Measure:Model:Dataset	6	0.010276	0.001713	3.444508	0.002137
Measure:Technique:Dataset	2	0.001038	0.000519	1.043918	0.352153
Model:Technique:Dataset	3	0.00744	0.00248	4.987578	0.001868
Measure:Model:Technique:Dataset	6	0.000413	6.89E-05	0.138575	0.991196
Residuals	4752	2.362765	0.000497		

The influence of the presence of label dependence on the relative performances of the SVM and XGBoost are studied in the next subsection.

Comparing performance of techniques with respect to label dependence (correlation)

Datasets 10 and 12 are compared in order to study the influence of label dependence on the performances of the two classifiers (see Table 5.33). There is no label dependence present for Dataset 10, while some degree of label dependence is present in Dataset 12. There is no clear expectation as to what the effect will be on the performance of the two classifiers if label dependence is introduced.

Table 5.33: Structure of Dataset 10 and Dataset 12.

	k	$p - k$	q	ρ	**Signal**	**Vector of Densities**	**Number of training instances**	**Number of test instances**
Dataset 10	50	10	6	0	10	0.4	30	10 000
Dataset 12	50	10	6	0.4	10	0.4	30	10 000

The results presented in Figures 5.31 to 5.34 are not as clear-cut as those in the previous two subsections. When one considers the Hamming-loss, Precision, and Recall, the performances improve if the correlation coefficient or IG is used as relevance measure, but label dependence negatively impacts the performance if ReliefF is used as a relevance measure. For One-error, the performances are negatively influenced by the presence of label dependence, irrespective of the relevance measure.

Once again, the differences observed between the two classifiers, the three relevance measures, the four different models, and the two datasets are all significant, as can be seen in Table 5.36.

The associated boxplots and results from the four-way ANOVA for other pairs of datasets that were used to investigate the influence of label dependence on the performance of the classifiers can be found in Appendices H and I. For example, the structure of Datasets 1 and 2 is summarised in Table 5.34:

Table 5.34: Structure of Dataset 1 and Dataset 2.

	k	$p - k$	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 2	10	10	6	0.4	10	0.4	80	10 000

Two other datasets that were compared in this part of the study are numbers 17 and 19. The structure of these datasets is summarised in Table 5.35:

Table 5.35: Structure of Dataset 17 and Dataset 19.

	k	$p - k$	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 17	50	10	6	0	10	vary	240	10 000
Dataset 19	50	10	6	0.4	10	vary	240	10 000

The results obtained from these two comparisons are however not consistent with those obtained from the comparison between Dataset 10 and Dataset 12. For Hamming-loss and One-error, the performance of all the procedures improves in the presence of label dependence. For the procedures based on the XGBoost classifier, all performances deteriorate if Precision and Recall are used as evaluation metrics. This inconsistence could possibly be explained by the influence of the number of training instances. For Dataset 10 and 12, $N = 30$, for Dataset 1 and 2, $N = 80$, and for Dataset 17 and 19, $N = 240$. The number of training instances seems to have a confounding influence on the results.

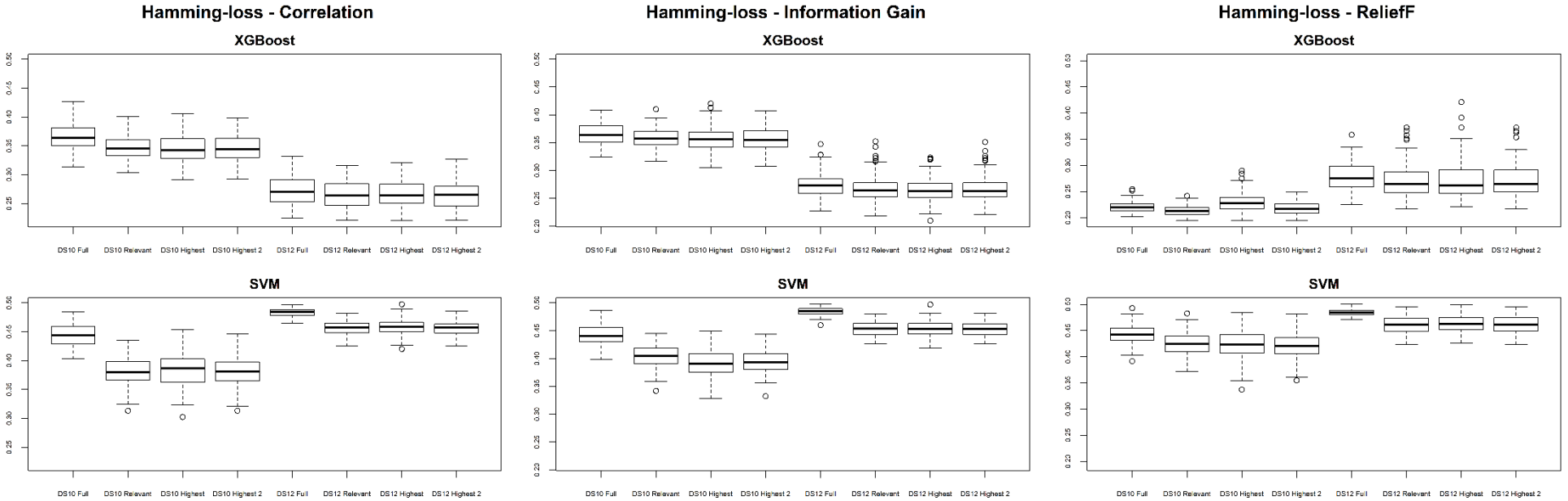


Figure 5.31 Hamming-loss: Dataset 10 vs Dataset 12.

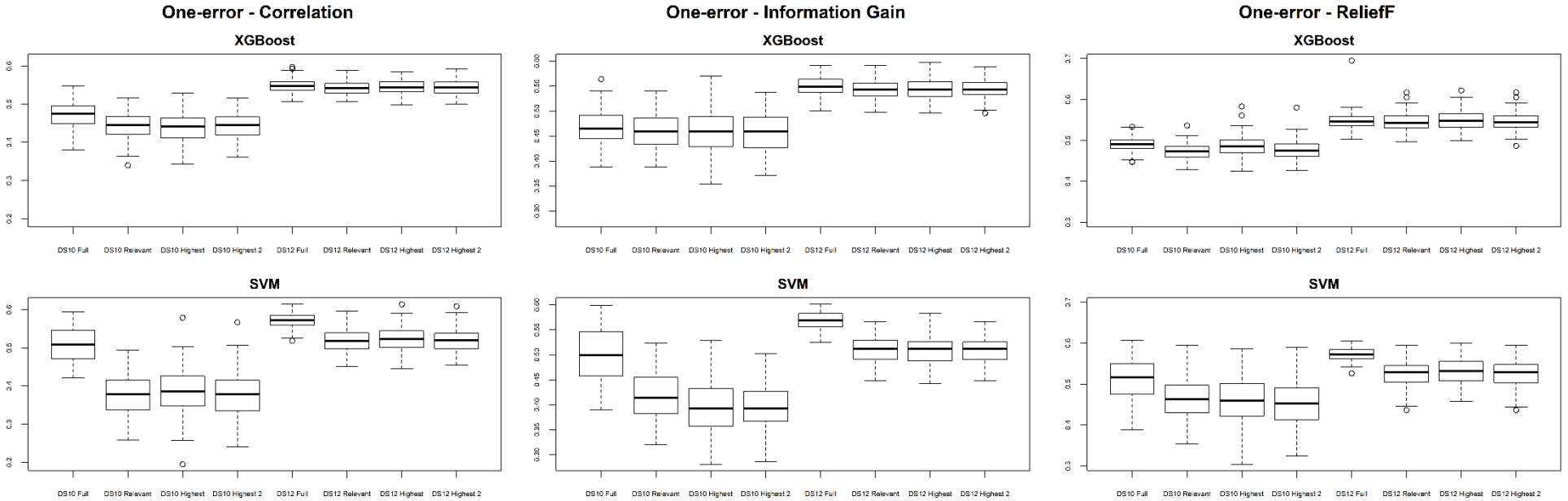


Figure 5.32 One-error: Dataset 10 vs Dataset 12.

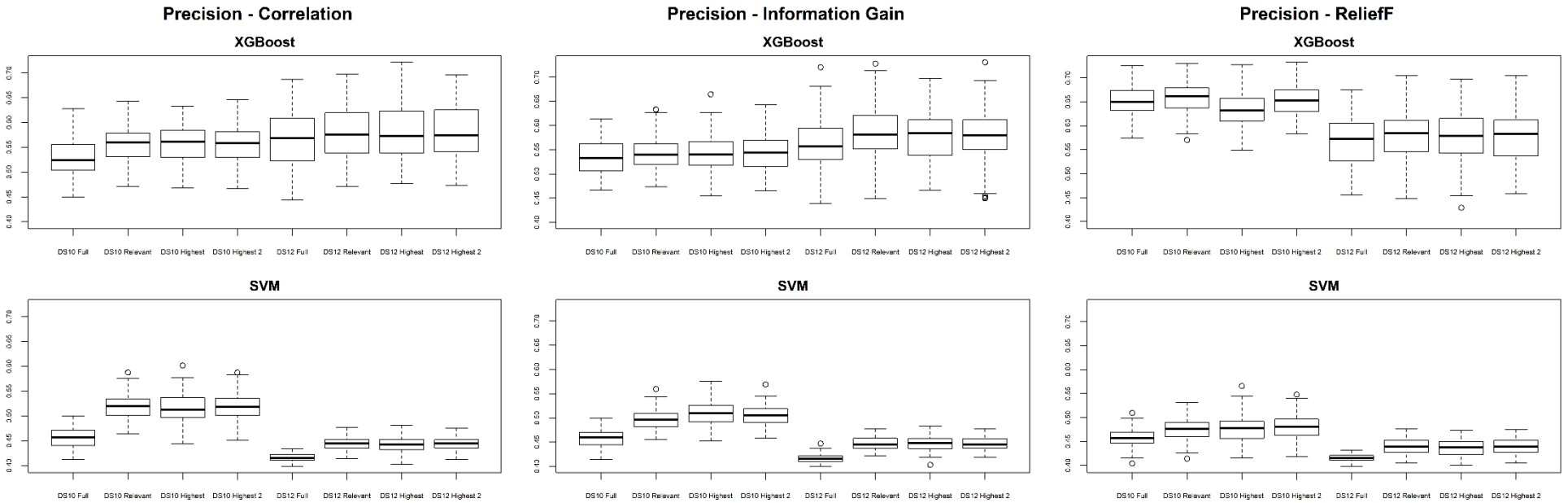


Figure 5.33 Precision: Dataset 10 vs Dataset 12.

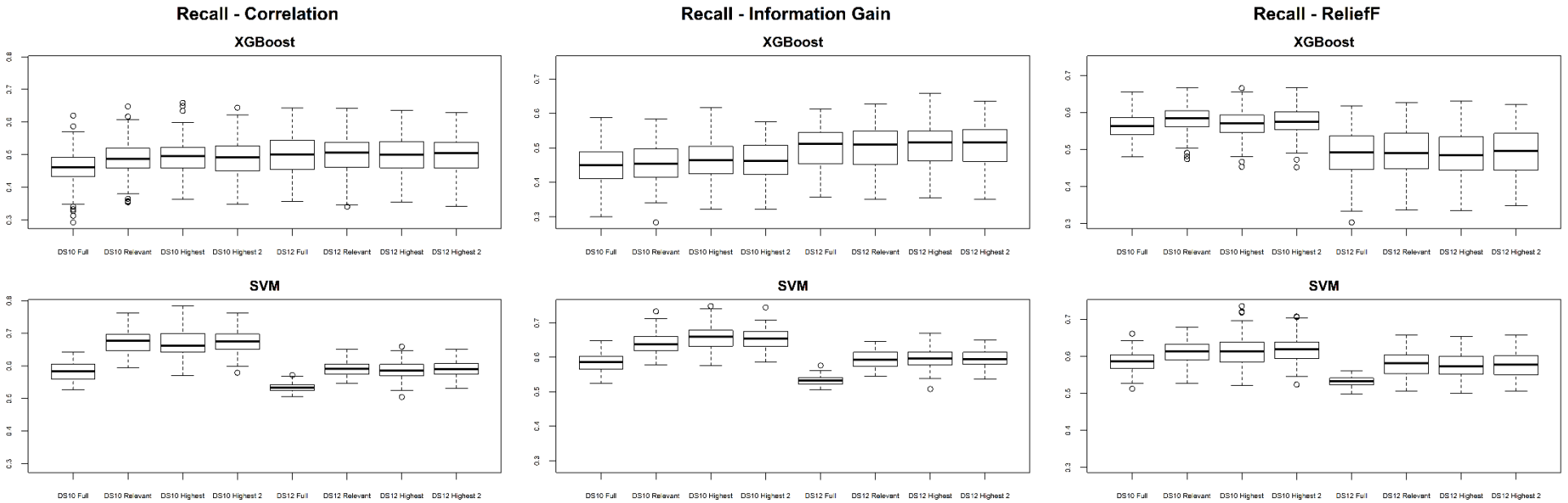


Figure 5.34 Recall: Dataset 10 vs Dataset 12.

Table 5.36: Four-way ANOVA: Dataset 10 vs Dataset 12.

Hamming-loss

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.143065	0.071532	148.1522	3.84E-63
Model	3	0.412797	0.137599	284.984	4.1E-170
Technique	1	18.31563	18.31563	37933.83	0
Dataset	1	0.370693	0.370693	767.7496	8.9E-157
Measure:Model	6	0.050977	0.008496	17.59655	2.93E-20
Measure:Technique	2	0.020963	0.010482	21.70886	4.12E-10
Model:Technique	3	0.170698	0.056899	117.845	1.3E-73
Measure:Dataset	2	0.061872	0.030936	64.07238	3.49E-28
Model:Dataset	3	0.017065	0.005688	11.78118	1.1E-07
Technique:Dataset	1	6.038925	6.038925	12507.32	0
Measure:Model:Technique	6	0.010222	0.001704	3.528549	0.001736
Measure:Model:Dataset	6	0.034998	0.005833	12.08094	1.61E-13
Measure:Technique:Dataset	2	0.020569	0.010284	21.29996	6.18E-10
Model:Technique:Dataset	3	0.014485	0.004828	9.999921	1.44E-06
Measure:Model:Technique:Dataset	6	0.005625	0.000938	1.941673	0.070482
Residuals	4752	2.294413	0.000483		

One-error

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.416328	0.208164	151.0317	2.56E-64
Model	3	1.438868	0.479623	347.9859	2.8E-204
Technique	1	0.411713	0.411713	298.7144	5.86E-65
Dataset	1	9.861025	9.861025	7154.579	0
Measure:Model	6	0.155905	0.025984	18.85257	8.4E-22
Measure:Technique	2	0.155252	0.077626	56.32076	6.69E-25
Model:Technique	3	0.976144	0.325381	236.0776	8.8E-143
Measure:Dataset	2	0.25659	0.128295	93.08327	2.22E-40
Model:Dataset	3	0.13044	0.04348	31.54655	3.48E-20
Technique:Dataset	1	0.021219	0.021219	15.39515	8.84E-05
Measure:Model:Technique	6	0.032188	0.005365	3.892345	0.0007
Measure:Model:Dataset	6	0.101975	0.016996	12.33116	8.01E-14
Measure:Technique:Dataset	2	0.071644	0.035822	25.99035	5.94E-12
Model:Technique:Dataset	3	0.066243	0.022081	16.02074	2.32E-10
Measure:Model:Technique:Dataset	6	0.018907	0.003151	2.286318	0.033124
Residuals	4752	6.549594	0.001378		

Precision

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.157047	0.078523	60.53223	1.1E-26
Model	3	0.505837	0.168612	129.9802	6.34E-81
Technique	1	11.0048	11.0048	8483.408	0
Dataset	1	0.095317	0.095317	73.47813	1.36E-17
Measure:Model	6	0.070497	0.01175	9.057521	7.18E-10
Measure:Technique	2	0.013557	0.006778	5.225336	0.00541
Model:Technique	3	0.120712	0.040237	31.01823	7.51E-20
Measure:Dataset	2	0.101606	0.050803	39.16325	1.35E-17
Model:Dataset	3	0.009553	0.003184	2.45466	0.061293
Technique:Dataset	1	2.362238	2.362238	1821.008	0
Measure:Model:Technique	6	0.006969	0.001161	0.895351	0.497143
Measure:Model:Dataset	6	0.046143	0.007691	5.928525	3.52E-06
Measure:Technique:Dataset	2	0.006075	0.003038	2.341733	0.096272
Model:Technique:Dataset	3	0.024908	0.008303	6.400372	0.000253
Measure:Model:Technique:Dataset	6	0.004509	0.000751	0.579265	0.747189
Residuals	4752	6.164363	0.001297		

Recall

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.262174	0.131087	57.42525	2.28E-25
Model	3	1.025521	0.34184	149.75	9.91E-93
Technique	1	16.8208	16.8208	7368.686	0
Dataset	1	0.218247	0.218247	95.60736	2.27E-22
Measure:Model	6	0.123706	0.020618	9.03199	7.7E-10
Measure:Technique	2	0.09881	0.049405	21.64276	4.4E-10
Model:Technique	3	0.562894	0.187631	82.19572	7.69E-52
Measure:Dataset	2	0.145918	0.072959	31.96111	1.63E-14
Model:Dataset	3	0.019922	0.006641	2.909027	0.033255
Technique:Dataset	1	2.197473	2.197473	962.647	1.3E-192
Measure:Model:Technique	6	0.021287	0.003548	1.554171	0.156375
Measure:Model:Dataset	6	0.074272	0.012379	5.422746	1.34E-05
Measure:Technique:Dataset	2	0.091433	0.045716	20.02701	2.18E-09
Model:Technique:Dataset	3	0.002437	0.000812	0.355837	0.784924
Measure:Model:Technique:Dataset	6	0.006552	0.001092	0.478406	0.824882
Residuals	4752	10.84758	0.002283		

In the last subsection the performances of the procedures are investigated for datasets exhibiting different vectors of densities.

Comparing performance of techniques with respect to different density vectors

Datasets 9 and 17 are identical except for their vectors of densities. For Dataset 9, the vector of densities is constant, namely $D = [0.4 \quad 0.4 \quad 0.4 \quad 0.4 \quad 0.4 \quad 0.4]$, while for Dataset 17 the entries in the vector of densities vary, namely $D = [0.25 \quad 0.31 \quad 0.2 \quad 0.42 \quad 0.28 \quad 0.35]$. Refer to Table 5.37. These density values were randomly selected at the start of the study and then used for all datasets where the densities were allowed to vary. Comparing these two datasets allows the study of the sensitivity of the procedures to changes in the vector of densities. Due to the sensitivity of the SVM classifier to smaller density values, one would expect the performance of the procedures based on the SVM to be influenced more severely than the performance of those procedures based on the XGBoost classifier.

Table 5.37: Structure of Dataset 9 and Dataset 17.

	k	p - k	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 9	50	10	6	0	10	0.4	240	10 000
Dataset 17	50	10	6	0	10	vary	240	10 000

For Hamming-loss (Figure 5.35) there is a small improvement in the performances of the procedures based on XGBoost when the vector of density values are varied, but the performances of the procedures based on the SVM are weaker for Dataset 17 than for Dataset 9. Based on Figure 5.36 one can conclude that the performances of all the procedures (irrespective of classifier) are negatively influenced by the varying density values in Dataset 17. For Precision and Recall, Figures 5.37 and 5.38, respectively, show that the performances of the procedures based on the SVM improve for Recall but deteriorate for Precision. The reverse holds for the procedures based on XGBoost.

When the four-way ANOVA in Table 5.39 is considered, it is important to note that not all observed differences are significant. More specifically, for the evaluation measures One-error

and Recall the differences observed between the relevance measures (the correlation coefficient, IG and ReliefF) are not significant.

In an attempt to further investigate the effect of label density, two other datasets, namely Datasets 16 and 24, were compared. The structure of these two datasets is summarised in Table 5.38.

Table 5.38: Structure of Dataset 16 and Dataset 24.

	k	$p - k$	q	ρ	**Signal**	**Vector of Densities**	**Number of training instances**	**Number of test instances**
Dataset 16	50	10	6	0.4	100	0.4	30	10 000
Dataset 24	50	10	6	0.4	100	Vary	30	10 000

For Hamming-loss, One-error, and Precision, the performances of the procedures worsen irrespective of classifier or relevance measure if constant densities are replaced by varying values. The differences are small but significant when considering the four-way ANOVA (see Appendix I). For Recall, the performances of the procedures based on the SVM classifier improve when the density vector is not kept constant at 0.4, but the performances of the procedures based on XGBoost are negatively influenced.

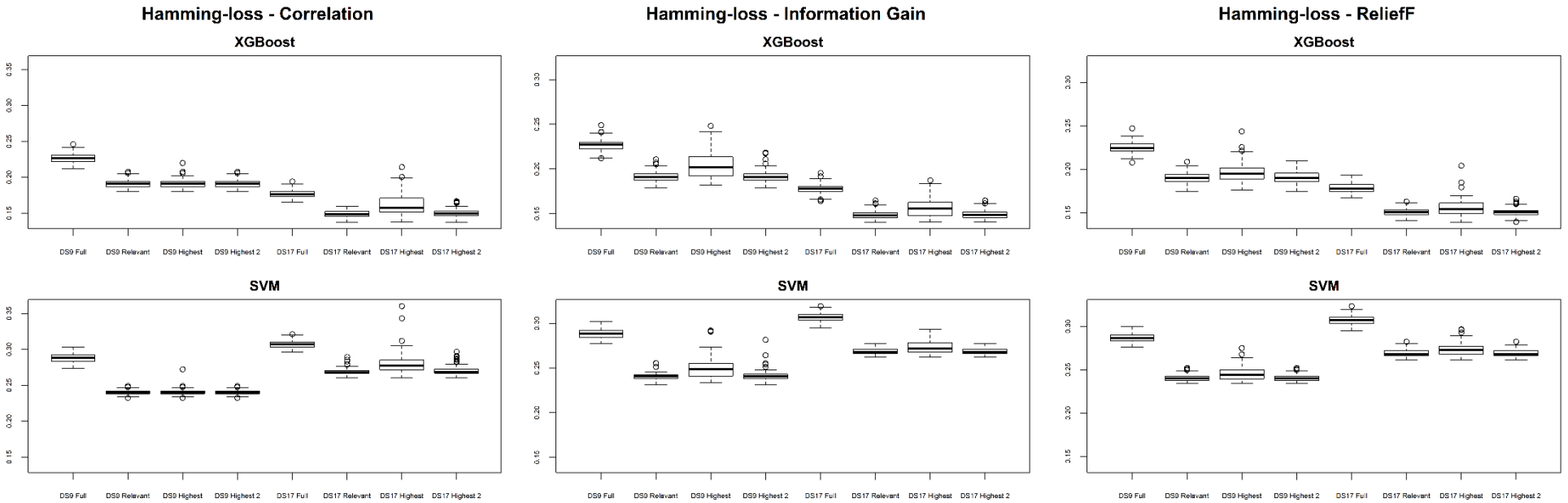


Figure 5.35 Hamming-loss: Dataset 9 vs Dataset 17.

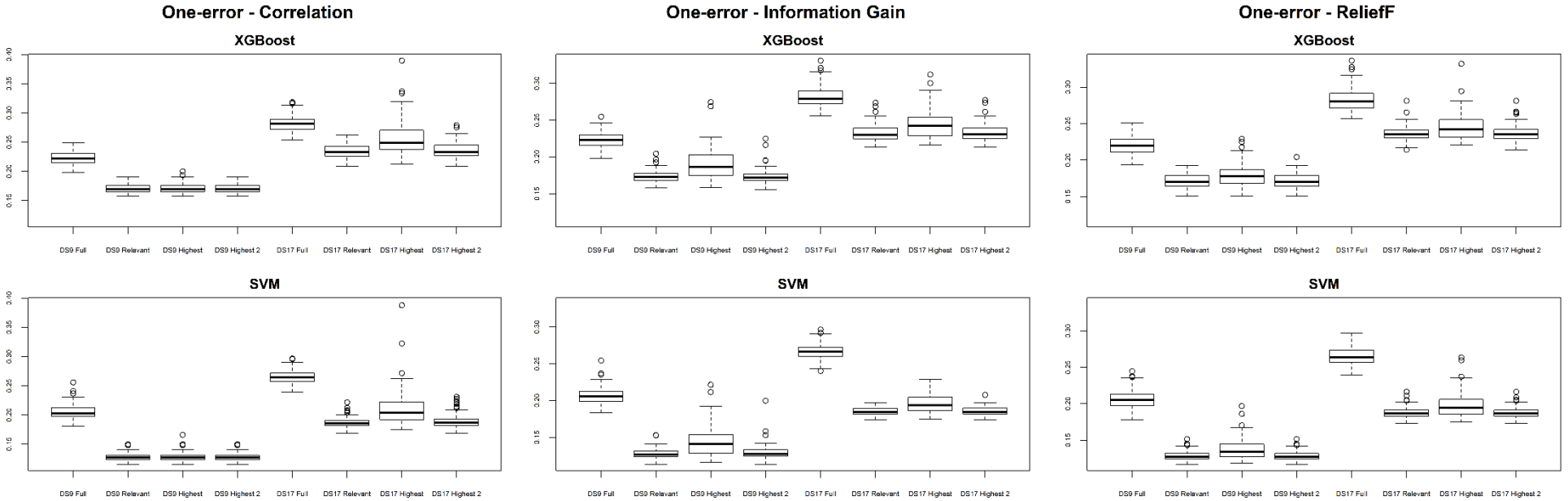


Figure 5.36 One-error: Dataset 9 vs Dataset 17.

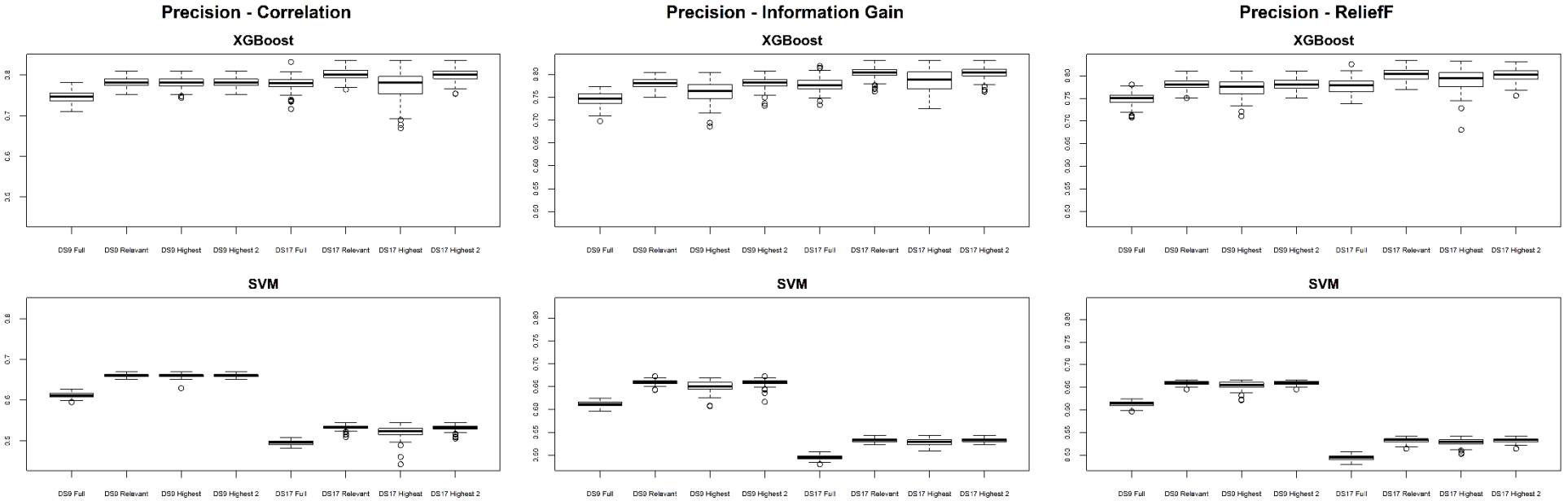


Figure 5.37 Precision: Dataset 9 vs Dataset 17.

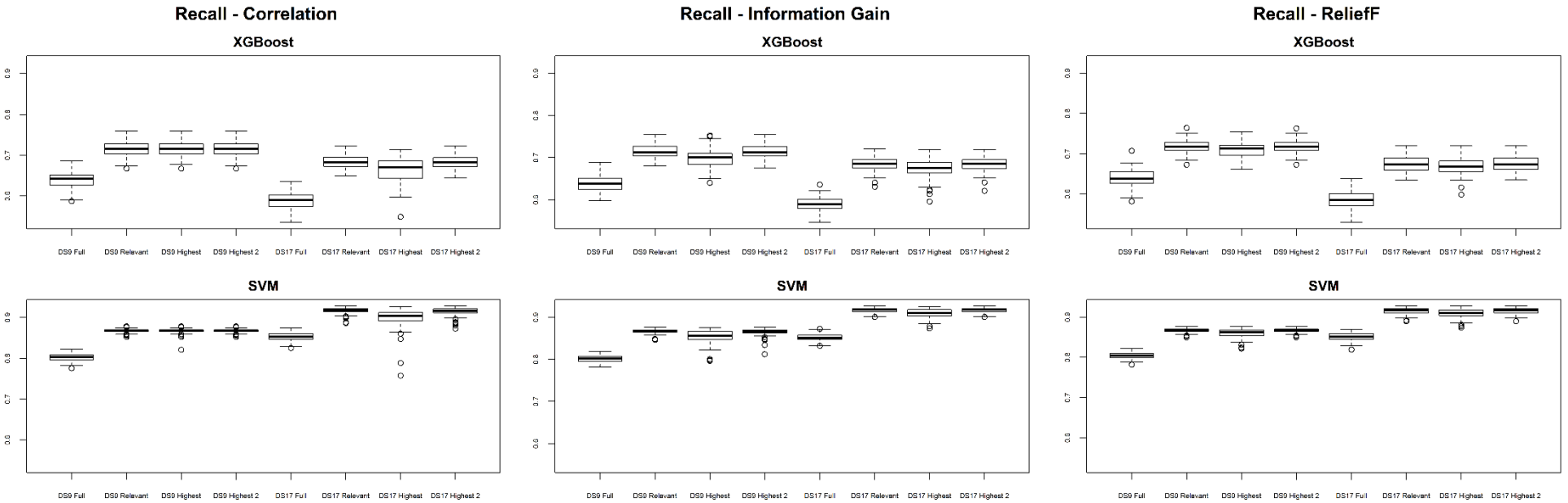


Figure 5.38 Recall: Dataset 9 vs Dataset 17.

Table 5.39: Four-way ANOVA: Dataset 9 vs Dataset 17.

Hamming-loss

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000714	0.000357	7.279461	0.000697
Model	3	1.133551	0.37785	7706.622	0
Technique	1	9.073454	9.073454	185061.9	0
Dataset	1	0.073936	0.073936	1507.999	9.3E-287
Measure:Model	6	0.00178	0.000297	6.051384	2.54E-06
Measure:Technique	2	4.34E-05	2.17E-05	0.44224	0.642622
Model:Technique	3	0.028766	0.009589	195.5682	1.4E-119
Measure:Dataset	2	0.005509	0.002755	56.18114	7.67E-25
Model:Dataset	3	0.018351	0.006117	124.7625	8.71E-78
Technique:Dataset	1	1.44651	1.44651	29502.97	0
Measure:Model:Technique	6	0.000199	3.32E-05	0.67637	0.668801
Measure:Model:Dataset	6	0.011386	0.001898	38.70453	3.64E-46
Measure:Technique:Dataset	2	0.000156	7.79E-05	1.588188	0.204404
Model:Technique:Dataset	3	0.00028	9.33E-05	1.902591	0.126882
Measure:Model:Technique:Dataset	6	8.72E-05	1.45E-05	0.296562	0.938797
Residuals	4752	0.232987	4.9E-05		

One-error

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000293	0.000146	0.872693	0.417892
Model	3	3.297327	1.099109	6552.396	0
Technique	1	1.747633	1.747633	10418.6	0
Dataset	1	4.624407	4.624407	27568.65	0
Measure:Model	6	0.003767	0.000628	3.742778	0.001019
Measure:Technique	2	0.000236	0.000118	0.704224	0.494544
Model:Technique	3	0.189414	0.063138	376.4005	2.5E-219
Measure:Dataset	2	0.019481	0.009741	58.06907	1.21E-25
Model:Dataset	3	0.009719	0.00324	19.31328	1.92E-12
Technique:Dataset	1	0.002925	0.002925	17.43476	3.03E-05
Measure:Model:Technique	6	0.001041	0.000173	1.034176	0.400772
Measure:Model:Dataset	6	0.032308	0.005385	32.10137	4.39E-38
Measure:Technique:Dataset	2	0.000903	0.000452	2.692093	0.067842
Model:Technique:Dataset	3	0.000264	8.81E-05	0.525355	0.664851
Measure:Model:Technique:Dataset	6	0.000119	1.99E-05	0.118552	0.994238
Residuals	4752	0.797108	0.000168		

Precision

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.002124	0.001062	6.424337	0.001636
Model	3	1.022735	0.340912	2062.225	0
Technique	1	46.50951	46.50951	281342.8	0
Dataset	1	3.243593	3.243593	19620.97	0
Measure:Model	6	0.003651	0.000608	3.680662	0.00119
Measure:Technique	2	0.000693	0.000347	2.096497	0.123
Model:Technique	3	0.071648	0.023883	144.4698	1.37E-89
Measure:Dataset	2	0.011686	0.005843	35.34453	5.8E-16
Model:Dataset	3	0.031868	0.010623	64.25876	1.01E-40
Technique:Dataset	1	6.434804	6.434804	38925.06	0
Measure:Model:Technique	6	0.001067	0.000178	1.075351	0.374665
Measure:Model:Dataset	6	0.022287	0.003715	22.46961	3E-26
Measure:Technique:Dataset	2	0.000847	0.000424	2.562804	0.077195
Model:Technique:Dataset	3	0.002901	0.000967	5.849215	0.000553
Measure:Model:Technique:Dataset	6	0.002364	0.000394	2.383701	0.026592
Residuals	4752	0.785565	0.000165		

Recall

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000652	0.000326	1.399751	0.24676
Model	3	4.652344	1.550781	6659.511	0
Technique	1	47.2833	47.2833	203048.4	0
Dataset	1	0.020295	0.020295	87.15163	1.51E-20
Measure:Model	6	0.003128	0.000521	2.238621	0.036852
Measure:Technique	2	0.001144	0.000572	2.456713	0.085825
Model:Technique	3	0.094066	0.031355	134.6486	1.01E-83
Measure:Dataset	2	0.015271	0.007636	32.79014	7.19E-15
Model:Dataset	3	0.015629	0.00521	22.3718	2.22E-14
Technique:Dataset	1	2.33616	2.33616	10032.16	0
Measure:Model:Technique	6	0.00037	6.17E-05	0.264787	0.953356
Measure:Model:Dataset	6	0.02481	0.004135	17.75693	1.86E-20
Measure:Technique:Dataset	2	0.003661	0.00183	7.860066	0.000391
Model:Technique:Dataset	3	0.017578	0.005859	25.1618	3.81E-16
Measure:Model:Technique:Dataset	6	0.001039	0.000173	0.743456	0.614612
Residuals	4752	1.106585	0.000233		

In the final section of this chapter, the performance of the RPFS procedures are compared to that of the two established multi-label FS techniques.

5.5 Comparison between feature selection approaches

In the previous sections the performance of the techniques that use RPFS to perform FS on the 24 synthetic datasets were considered. In this section a comparative study will be presented. The RPFS techniques will be compared with the methods proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013), as discussed in Sections 3.3.5 and 3.3.6 respectively. This comparative study has some limitations and the following important aspects should be noted:

1. The full models for Probe Selection (PS) and RPFS are not directly comparable. This is due to the thresholding applied using RPFS and the manner in which PS allocates labels. The allocation of labels by PS is influenced by the parameter *label-cut*. In Section 3.3.5 label-cut is defined as the minimum number of labels for which a feature should be deemed locally relevant in order to be deemed globally relevant. For the comparison study performed in this dissertation, the label-cut value is specified to be four. This decision is based on the results obtained by Sandrock and Steel (2016).
2. The comparison is only performed using the SVM classifier.
3. In order for a ranking of the techniques to be performed, the medians are used. This is motivated by the large number of outliers present in some datasets. Only the ranks of the medians are considered and therefore, by not considering the magnitudes of observed differences, some information is lost.
4. For a similar reason, the IQRs are used as a measure of variation when comparing the different techniques.
5. Boxplots that provide visual representations that support this section can be found in Appendix J[12].

The aim of Section 5.5 is to:

a) determine, based on the median, which FS procedure (Spolaôr, PS or RPFS) performs best for each of the four evaluation measures,

[12] All appendices can be found at: https://sites.google.com/view/ivonacontardo/appendices-to-aspects-of-multi-label-classification.

b) evaluate the variation (specifically the IQR) associated with each of these FS procedures,

c) compare the various FS techniques to the full model, and

d) compare the FS procedures with respect to the Feature Reduction as defined by Spolaôr *et al*. (2013).

In Figure 5.39 and 5.40, the three procedures are compared based on Dataset 1. It is clear from these figures that the results for the FS procedures based on PS and Spolaôr show much larger variation than those for the RPFS procedures for all four evaluation measures. The RPFS method also performs better than the full model, whereas this is not the case for the other two methods.

For Hamming-loss and One-error the medians for the RPFS procedures are all smaller than those for PS and Spolaôr, implying that the RPFS procedures perform better. For Precision in Figure 5.40, the medians of the RPFS Relevant and Highest 2 techniques (for all relevance measures) are similar to the median of PS Correlation. Finally, for Recall the RPFS procedures have larger medians than do the PS and Spolaôr approaches.

The graphs for the other 23 datasets are available in Appendix J. As before, these graphs do not allow for the efficient overall comparison of the techniques. In order to compare all 15 procedures in terms of all 24 datasets, the results of the Method of Pairwise Comparisons are provided next. The interquartile ranges (IQRs) are included to provide an insight into the amount of variation that is present in the results. A darker green indicates a larger IQR. In Table 5.40 to 5.43, the 15 techniques are ranked according to their median values per evaluation measure. The full models are shaded pink in order to compare the full models to the reduced models obtained when FS is applied.

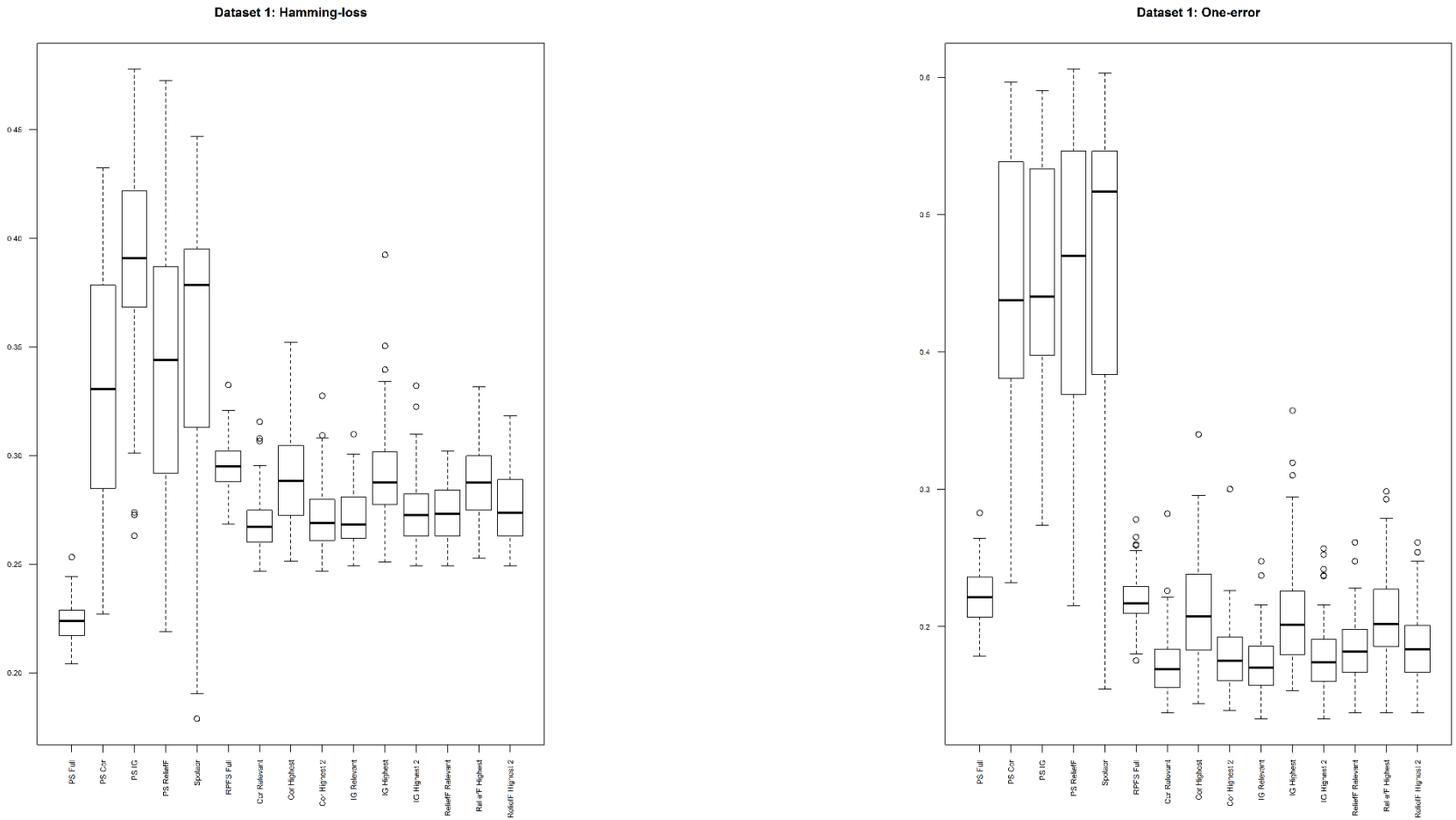


Figure 5.39 Comparing Hamming-loss and One-error for PS, RPFS, and Spolaôr FS procedures: Dataset 1.

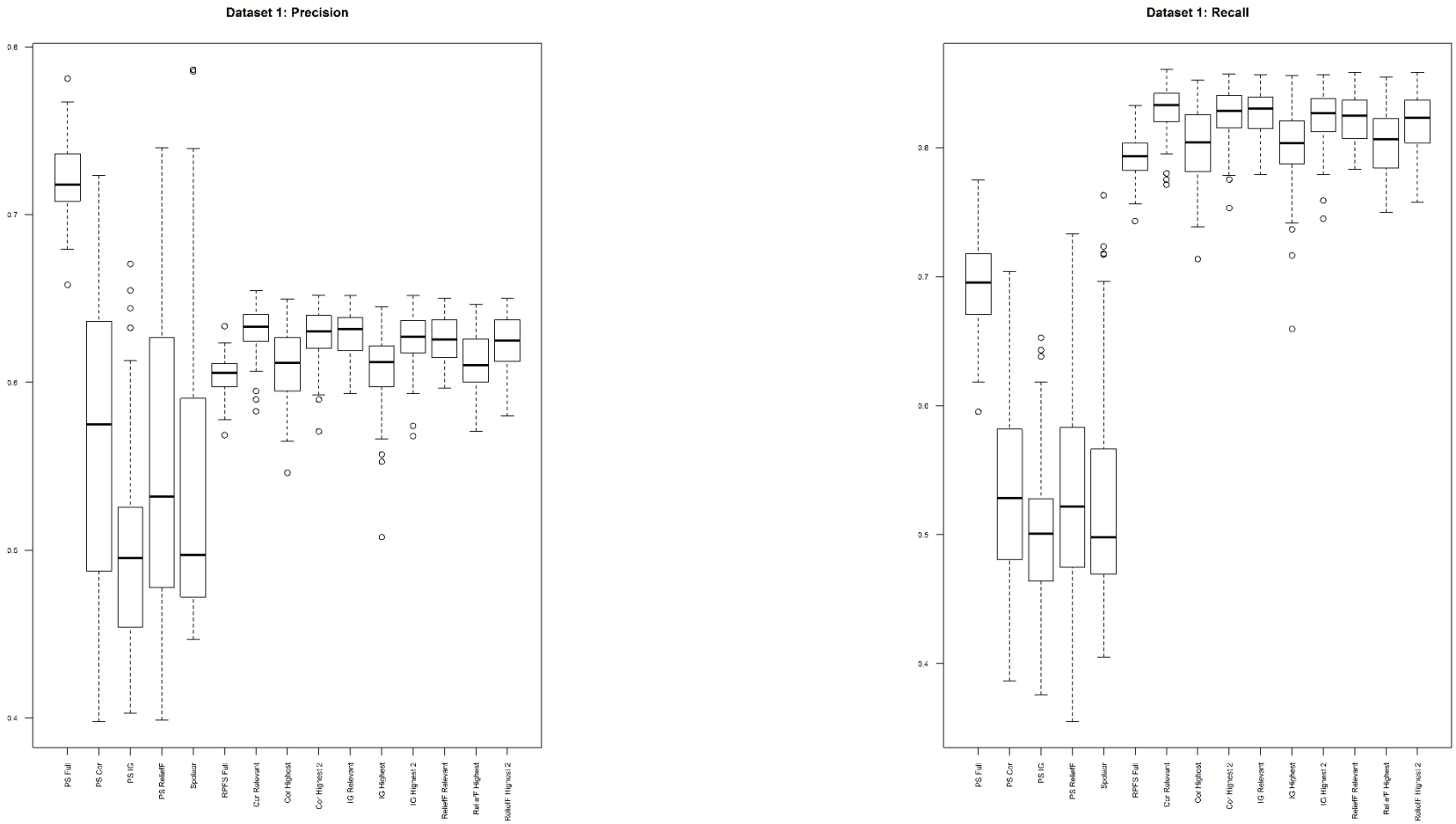


Figure 5.40 Comparing Precision and Recall for PS, RPFS, and Spolaôr FS procedures: Dataset 1.

Table 5.40: Hamming-loss.

	Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dataset 1	**Median**	PS Full	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Highest	Cor Highest	PS Cor	RPFS Full	PS ReliefF	Spolaor	PS IG
	IQR	0.0114	0.0146	0.0187	0.0184	0.0192	0.0211	0.0258	0.0246	0.0239	0.0321	0.0549	0.0138	0.1015	0.1002	0.0507
Dataset 2	**Median**	PS Full	PS Cor	PS ReliefF	Spolaor	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full	PS IG
	IQR	0.0017	0.0685	0.0413	0.0022	0.0084	0.0102	0.0095	0.0132	0.0107	0.0170	0.0171	0.0178	0.0246	0.0089	0.0023
Dataset 3	**Median**	Spolaor	PS Full	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full	PS Cor	PS ReliefF	PS IG
	IQR	0.0439	0.0134	0.0052	0.0052	0.0058	0.0049	0.0049	0.0057	0.0060	0.0066	0.0098	0.0093	0.1409	0.1300	0.1263
Dataset 4	**Median**	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full	IG Relevant	ReliefF Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0156	0.0156	0.0148	0.0149	0.0128	0.0042	0.0050	0.0056	0.0047	0.0071	0.0104	0.0079	0.0157	0.0214	0.0082
Dataset 5	**Median**	PS Full	Spolaor	PS Cor	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full	PS ReliefF	PS IG
	IQR	0.0116	0.0697	0.0790	0.0096	0.0096	0.0089	0.0100	0.0104	0.0140	0.0145	0.0157	0.0180	0.0113	0.0631	0.0728
Dataset 6	**Median**	Spolaor	PS Cor	PS Full	PS ReliefF	PS IG	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0154	0.0102	0.0122	0.0233	0.0579	0.0087	0.0086	0.0116	0.0108	0.0098	0.0131	0.0166	0.0149	0.0168	0.0077
Dataset 7	**Median**	Spolaor	PS Full	PS Cor	PS ReliefF	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full	PS IG	IG Relevant	IG Highest 2	IG Highest
	IQR	0.0166	0.0155	0.1378	0.0944	0.0061	0.0059	0.0057	0.0057	0.0061	0.0054	0.0126	0.1181	0.0296	0.0290	0.0356
Dataset 8	**Median**	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	ReliefF Highest 2	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full
	IQR	0.0161	0.0167	0.0159	0.0178	0.0140	0.0061	0.0060	0.0052	0.0076	0.0064	0.0065	0.0097	0.0085	0.0134	0.0062
Dataset 9	**Median**	PS Full	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full	PS Cor	PS ReliefF	Spolaor	PS IG
	IQR	0.0078	0.0046	0.0046	0.0035	0.0035	0.0035	0.0045	0.0049	0.0104	0.0141	0.0090	0.1031	0.0754	0.1877	0.0625
Dataset 10	**Median**	PS Full	PS Cor	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	IG Highest 2	Spolaor	PS ReliefF	IG Relevant	PS IG	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full
	IQR	0.0178	0.0569	0.0322	0.0329	0.0399	0.0331	0.0283	0.0445	0.0657	0.0279	0.0588	0.0306	0.0360	0.0290	0.0303
Dataset 11	**Median**	PS Cor	PS IG	PS ReliefF	Spolaor	PS Full	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0057	0.0069	0.0085	0.0081	0.0063	0.0040	0.0044	0.0038	0.0043	0.0106	0.0141	0.0156	0.0204	0.0198	0.0052
Dataset 12	**Median**	PS Cor	Spolaor	PS ReliefF	PS Full	PS IG	IG Highest	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0197	0.0181	0.0331	0.0175	0.0716	0.0187	0.0186	0.0194	0.0163	0.0153	0.0163	0.0245	0.0255	0.0229	0.0094
Dataset 13	**Median**	Spolaor	PS ReliefF	PS Full	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest	RPFS Full	PS Cor	PS IG
	IQR	0.1169	0.0958	0.0195	0.0368	0.0416	0.0347	0.0371	0.0343	0.0301	0.0312	0.0308	0.0508	0.0385	0.1075	0.0782
Dataset 14	**Median**	Cor Highest 2	Cor Relevant	Cor Highest	PS Full	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	Spolaor	PS Cor	PS ReliefF	PS IG	RPFS Full
	IQR	0.0308	0.0301	0.0312	0.0195	0.0343	0.0347	0.0371	0.0416	0.0368	0.0508	0.1169	0.1075	0.0958	0.0782	0.0385
Dataset 15	**Median**	PS ReliefF	Spolaor	PS Cor	PS IG	PS Full	Cor Relevant	IG Relevant	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0076	0.0076	0.0080	0.0074	0.0083	0.0027	0.0023	0.0027	0.0021	0.0027	0.0026	0.0054	0.0044	0.0047	0.0056
Dataset 16	**Median**	Spolaor	PS Cor	PS IG	PS ReliefF	PS Full	Cor Relevant	Cor Highest 2	IG Highest 2	IG Relevant	ReliefF Relevant	IG Highest	ReliefF Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0173	0.0143	0.0237	0.0181	0.0139	0.0195	0.0252	0.0233	0.0209	0.0199	0.0316	0.0224	0.0289	0.0278	0.0179
Dataset 17	**Median**	PS Full	Spolaor	Cor Relevant	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	PS Cor	Cor Highest	PS ReliefF	RPFS Full	PS IG
	IQR	0.0076	0.1014	0.0043	0.0050	0.0051	0.0061	0.0060	0.0056	0.0106	0.0090	0.0968	0.0132	0.0623	0.0065	0.0505
Dataset 18	**Median**	PS Full	PS Cor	Spolaor	PS ReliefF	PS IG	Cor Highest	Cor Highest 2	IG Highest 2	IG Relevant	Cor Relevant	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0148	0.0430	0.0603	0.0503	0.0598	0.0310	0.0264	0.0290	0.0287	0.0305	0.0356	0.0272	0.0273	0.0259	0.0221
Dataset 19	**Median**	PS Cor	Spolaor	PS IG	PS ReliefF	PS Full	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0097	0.0094	0.0091	0.0107	0.0073	0.0034	0.0042	0.0036	0.0036	0.0047	0.0049	0.0065	0.0057	0.0096	0.0053
Dataset 20	**Median**	PS Cor	Spolaor	PS ReliefF	PS Full	PS IG	IG Relevant	IG Highest 2	IG Highest	Cor Highest 2	Cor Relevant	Cor Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0296	0.0329	0.0415	0.0198	0.0709	0.0156	0.0174	0.0165	0.0160	0.0130	0.0188	0.0163	0.0150	0.0156	0.0149
Dataset 21	**Median**	Spolaor	PS ReliefF	PS Full	PS Cor	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest	RPFS Full	PS IG
	IQR	0.0100	0.0194	0.0077	0.1391	0.0028	0.0028	0.0028	0.0022	0.0022	0.0027	0.0027	0.0028	0.0022	0.0074	0.1104
Dataset 22	**Median**	Spolaor	PS Full	PS Cor	PS ReliefF	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	PS IG	RPFS Full
	IQR	0.0540	0.0246	0.1007	0.0953	0.0197	0.0216	0.0260	0.0237	0.0235	0.0273	0.0281	0.0266	0.0259	0.0878	0.0298
Dataset 23	**Median**	PS Cor	Spolaor	PS ReliefF	PS IG	PS Full	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0102	0.0105	0.0105	0.0087	0.0075	0.0026	0.0026	0.0029	0.0026	0.0026	0.0037	0.0070	0.0162	0.0245	0.0042
Dataset 24	**Median**	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest	Cor Highest 2	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0241	0.0212	0.0312	0.0413	0.0185	0.0207	0.0216	0.0160	0.0190	0.0201	0.0177	0.0230	0.0211	0.0227	0.0181

Table 5.41: One-error.

	Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dataset 1	Median	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full	PS Full	PS Cor	PS IG	Spolaor	PS ReliefF
	IQR	0.0276	0.0281	0.0303	0.0316	0.0314	0.0342	0.0459	0.0414	0.0541	0.0192	0.0288	0.0920	0.0831	0.2179	0.1936
Dataset 2	Median	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	PS Full	ReliefF Highest	RPFS Full	PS Cor	PS ReliefF	PS IG	Spolaor
	IQR	0.0189	0.0230	0.0200	0.0296	0.0224	0.0355	0.0321	0.0444	0.0057	0.0418	0.0235	0.0252	0.0317	0.0070	0.0061
Dataset 3	Median	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Spolaor	PS Full	RPFS Full	PS IG	PS Cor	PS ReliefF
	IQR	0.0041	0.0044	0.0041	0.0037	0.0037	0.0038	0.0046	0.0043	0.0063	0.0127	0.0126	0.0151	0.1649	0.2561	0.2592
Dataset 4	Median	IG Relevant	ReliefF Relevant	IG Highest 2	Cor Relevant	PS Cor	Spolaor	Cor Highest 2	ReliefF Highest 2	PS ReliefF	IG Highest	PS IG	Cor Highest	ReliefF Highest	RPFS Full	PS Full
	IQR	0.0144	0.0132	0.0171	0.0183	0.0158	0.0164	0.0212	0.0292	0.0151	0.0242	0.0201	0.0370	0.0377	0.0229	0.0199
Dataset 5	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full	PS Full	Spolaor	PS Cor	PS IG	PS ReliefF
	IQR	0.0228	0.0220	0.0263	0.0215	0.0211	0.0269	0.0257	0.0298	0.0360	0.0259	0.0301	0.1379	0.0946	0.1036	0.0856
Dataset 6	Median	Cor Relevant	IG Relevant	Cor Highest 2	Spolaor	IG Highest 2	ReliefF Relevant	PS Cor	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	PS Full	RPFS Full	PS IG	PS ReliefF
	IQR	0.0195	0.0214	0.0310	0.0332	0.0308	0.0234	0.0408	0.0277	0.0482	0.0360	0.0321	0.0267	0.0246	0.0300	0.0705
Dataset 7	Median	ReliefF Relevant	ReliefF Highest 2	Cor Highest	Cor Relevant	Cor Highest 2	ReliefF Highest	Spolaor	RPFS Full	PS Full	IG Relevant	IG Highest 2	IG Highest	PS IG	PS Cor	PS ReliefF
	IQR	0.0065	0.0065	0.0064	0.0065	0.0065	0.0071	0.0099	0.0124	0.0155	0.0662	0.0692	0.0899	0.2061	0.2114	0.1806
Dataset 8	Median	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	ReliefF Highest 2	Spolaor	Cor Highest 2	PS Cor	ReliefF Highest	PS ReliefF	Cor Highest	PS IG	PS Full	RPFS Full
	IQR	0.0134	0.0133	0.0110	0.0143	0.0108	0.0149	0.0115	0.0150	0.0127	0.0186	0.0105	0.0235	0.0190	0.0165	0.0210
Dataset 9	Median	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	IG Highest	RPFS Full	PS Full	PS Cor	PS IG	PS ReliefF	Spolaor
	IQR	0.0075	0.0085	0.0063	0.0068	0.0063	0.0070	0.0070	0.0169	0.0257	0.0147	0.0157	0.2068	0.1428	0.1450	0.3439
Dataset 10	Median	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	IG Highest	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full	PS Full	PS Cor	Spolaor	PS ReliefF	PS IG
	IQR	0.0768	0.0803	0.0778	0.0596	0.0739	0.0713	0.0765	0.0775	0.0665	0.0737	0.0717	0.0644	0.0628	0.0428	0.0507
Dataset 11	Median	PS Cor	IG Relevant	Cor Relevant	ReliefF Relevant	Cor Highest 2	PS ReliefF	Cor Highest	PS IG	IG Highest 2	Spolaor	ReliefF Highest 2	IG Highest	ReliefF Highest	PS Full	RPFS Full
	IQR	0.0095	0.0106	0.0119	0.0101	0.0132	0.0184	0.0235	0.0225	0.0289	0.0314	0.0349	0.0379	0.0358	0.0124	0.0159
Dataset 12	Median	IG Highest 2	IG Highest	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	PS Cor	Spolaor	PS IG	PS Full	RPFS Full	PS ReliefF
	IQR	0.0353	0.0370	0.0373	0.0419	0.0398	0.0439	0.0428	0.0402	0.0479	0.0591	0.0539	0.0446	0.0205	0.0254	0.0344
Dataset 13	Median	ReliefF Relevant	ReliefF Highest 2	Spolaor	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	PS ReliefF	PS Full	RPFS Full	PS IG	PS Cor
	IQR	0.0031	0.0031	0.0027	0.0032	0.0032	0.0032	0.0020	0.0020	0.0020	0.0042	0.0180	0.0057	0.0082	0.2163	0.1999
Dataset 14	Median	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	PS Full	RPFS Full	Spolaor	PS IG	PS ReliefF	PS Cor
	IQR	0.0680	0.0603	0.0643	0.0692	0.0588	0.0825	0.0639	0.0704	0.0850	0.1056	0.1289	0.2249	0.1904	0.1283	0.1448
Dataset 15	Median	Cor Relevant	ReliefF Relevant	Cor Highest 2	IG Relevant	ReliefF Highest 2	IG Highest 2	Spolaor	PS ReliefF	Cor Highest	PS Cor	ReliefF Highest	IG Highest	PS IG	PS Full	RPFS Full
	IQR	0.0065	0.0060	0.0070	0.0062	0.0084	0.0063	0.0093	0.0093	0.0139	0.0098	0.0140	0.0119	0.0085	0.0121	0.0115
Dataset 16	Median	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	IG Highest	Cor Highest	ReliefF Highest 2	Spolaor	ReliefF Relevant	ReliefF Highest	PS Cor	PS IG	PS ReliefF	RPFS Full	PS Full
	IQR	0.0523	0.0579	0.0634	0.0606	0.0658	0.0693	0.0515	0.0613	0.0487	0.0524	0.0914	0.0931	0.1011	0.0400	0.0518
Dataset 17	Median	IG Relevant	IG Highest 2	Cor Relevant	ReliefF Highest 2	ReliefF Relevant	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	PS Full	RPFS Full	Spolaor	PS IG	PS Cor	PS ReliefF
	IQR	0.0077	0.0080	0.0089	0.0083	0.0086	0.0109	0.0179	0.0197	0.0293	0.0172	0.0143	0.1628	0.1089	0.1235	0.1011
Dataset 18	Median	Cor Highest 2	Cor Highest	Cor Relevant	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full	PS Full	PS Cor	Spolaor	PS IG	PS ReliefF
	IQR	0.0775	0.0879	0.0764	0.0703	0.0728	0.0756	0.0575	0.0711	0.0554	0.0624	0.0550	0.0518	0.0429	0.0616	0.0626
Dataset 19	Median	PS Cor	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	Spolaor	PS ReliefF	PS IG	PS Full	RPFS Full
	IQR	0.0116	0.0122	0.0132	0.0124	0.0124	0.0110	0.0136	0.0157	0.0146	0.0213	0.0273	0.0318	0.0341	0.0131	0.0140
Dataset 20	Median	IG Relevant	IG Highest 2	IG Highest	Cor Highest 2	Cor Highest	Cor Relevant	Spolaor	PS Cor	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	PS Full	PS IG	PS ReliefF	RPFS Full
	IQR	0.0389	0.0412	0.0468	0.0368	0.0443	0.0282	0.0302	0.0272	0.0268	0.0286	0.0294	0.0371	0.0576	0.0653	0.0392
Dataset 21	Median	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	IG Highest	Spolaor	PS ReliefF	RPFS Full	PS Full	PS IG	PS Cor
	IQR	0.0044	0.0044	0.0044	0.0035	0.0035	0.0035	0.0039	0.0039	0.0040	0.0050	0.0188	0.0060	0.0054	0.1582	0.2104
Dataset 22	Median	Cor Highest 2	Cor Relevant	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	Spolaor	PS Full	RPFS Full	PS IG	PS Cor	PS ReliefF
	IQR	0.0418	0.0481	0.0461	0.0467	0.0484	0.0585	0.0657	0.0643	0.0696	0.0977	0.1131	0.1058	0.1512	0.1056	0.1287
Dataset 23	Median	ReliefF Relevant	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest 2	Spolaor	PS Cor	PS ReliefF	Cor Highest	PS IG	ReliefF Highest 2	PS Full	RPFS Full	ReliefF Highest
	IQR	0.0069	0.0060	0.0060	0.0077	0.0062	0.0100	0.0086	0.0086	0.0089	0.0163	0.0098	0.0370	0.0144	0.0132	0.0670
Dataset 24	Median	IG Relevant	Cor Relevant	IG Highest 2	Cor Highest 2	IG Highest	Cor Highest	Spolaor	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	PS Cor	PS IG	PS ReliefF	PS Full	RPFS Full
	IQR	0.0456	0.0549	0.0461	0.0612	0.0452	0.0646	0.0655	0.0538	0.0575	0.0692	0.1050	0.0872	0.1224	0.0353	0.0465

Table 5.42: Precision.

	Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dataset 1	Median	PS Full	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	PS Cor	IG Highest	Cor Highest	ReliefF Highest	RPFS Full	PS ReliefF	Spolaor	PS IG
	IQR	0.0282	0.0159	0.0193	0.0196	0.0189	0.0221	0.0244	0.0772	0.0242	0.0318	0.0256	0.0137	0.1566	0.1537	0.0653
Dataset 2	Median	PS Full	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	PS Cor	RPFS Full	PS ReliefF	Spolaor	PS IG
	IQR	0.0064	0.0098	0.0086	0.0103	0.0131	0.0121	0.0164	0.0181	0.0177	0.0215	0.0438	0.0110	0.0294	0.0057	0.0062
Dataset 3	Median	Spolaor	PS Full	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	IG Highest	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full	PS Cor	PS ReliefF	PS IG
	IQR	0.0775	0.0304	0.0057	0.0057	0.0061	0.0061	0.0071	0.0068	0.0053	0.0064	0.0082	0.0104	0.2170	0.2046	0.1599
Dataset 4	Median	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full	IG Relevant	ReliefF Relevant	Cor Relevant	IG Highest 2	Cor Highest 2	ReliefF Highest 2	IG Highest	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0525	0.0525	0.0513	0.0521	0.0630	0.0072	0.0062	0.0077	0.0081	0.0093	0.0119	0.0108	0.0173	0.0187	0.0085
Dataset 5	Median	PS Full	Spolaor	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Relevant	IG Highest	PS Cor	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full	PS IG	PS ReliefF
	IQR	0.0385	0.1566	0.0101	0.0103	0.0110	0.0104	0.0119	0.0141	0.1598	0.0138	0.0158	0.0181	0.0115	0.1207	0.1278
Dataset 6	Median	Spolaor	PS Cor	PS Full	PS ReliefF	PS IG	IG Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0474	0.0498	0.0445	0.0763	0.1176	0.0096	0.0092	0.0104	0.0119	0.0103	0.0130	0.0170	0.0178	0.0176	0.0087
Dataset 7	Median	Spolaor	PS Full	PS Cor	PS ReliefF	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	PS IG	RPFS Full	IG Relevant	IG Highest 2	IG Highest
	IQR	0.0370	0.0402	0.3171	0.2236	0.0065	0.0064	0.0065	0.0064	0.0064	0.0067	0.1978	0.0135	0.0298	0.0293	0.0364
Dataset 8	Median	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	IG Highest	ReliefF Highest	Cor Highest 2	Cor Highest	RPFS Full
	IQR	0.0533	0.0518	0.0551	0.0515	0.0559	0.0083	0.0085	0.0069	0.0087	0.0073	0.0103	0.0097	0.0099	0.0141	0.0068
Dataset 9	Median	PS Full	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full	PS Cor	PS ReliefF	Spolaor	PS IG
	IQR	0.0188	0.0058	0.0058	0.0047	0.0047	0.0047	0.0061	0.0065	0.0111	0.0150	0.0085	0.1755	0.1292	0.2619	0.0846
Dataset 10	Median	PS Full	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	IG Highest 2	PS Cor	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	Spolaor	PS ReliefF	PS IG	RPFS Full
	IQR	0.0308	0.0319	0.0339	0.0393	0.0338	0.0275	0.0928	0.0266	0.0330	0.0355	0.0292	0.0424	0.0502	0.0750	0.0308
Dataset 11	Median	PS Cor	PS ReliefF	PS IG	Spolaor	PS Full	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0281	0.0270	0.0328	0.0290	0.0294	0.0067	0.0059	0.0056	0.0066	0.0109	0.0167	0.0178	0.0235	0.0225	0.0085
Dataset 12	Median	PS Cor	Spolaor	PS ReliefF	PS IG	PS Full	IG Highest	IG Highest 2	IG Relevant	Cor Highest 2	Cor Relevant	Cor Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full
	IQR	0.0875	0.0880	0.0993	0.1038	0.0611	0.0203	0.0191	0.0207	0.0164	0.0171	0.0208	0.0255	0.0248	0.0266	0.0111
Dataset 13	Median	Spolaor	PS ReliefF	PS Full	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest	RPFS Full	PS Cor	PS IG
	IQR	0.0179	0.0358	0.0153	0.0044	0.0049	0.0044	0.0044	0.0044	0.0038	0.0038	0.0038	0.0053	0.0057	0.1763	0.1571
Dataset 14	Median	PS Full	Cor Highest 2	Cor Relevant	Cor Highest	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	Spolaor	ReliefF Highest	PS Cor	PS ReliefF	RPFS Full	PS IG
	IQR	0.0438	0.0304	0.0294	0.0315	0.0351	0.0354	0.0386	0.0404	0.0360	0.1736	0.0514	0.1638	0.1405	0.0394	0.0991
Dataset 15	Median	PS ReliefF	Spolaor	PS Cor	PS IG	PS Full	Cor Relevant	Cor Highest 2	ReliefF Relevant	IG Relevant	ReliefF Highest 2	IG Highest 2	Cor Highest	ReliefF Highest	IG Highest	RPFS Full
	IQR	0.0311	0.0311	0.0311	0.0324	0.0371	0.0049	0.0057	0.0053	0.0048	0.0055	0.0056	0.0068	0.0075	0.0067	0.0067
Dataset 16	Median	Spolaor	PS Cor	PS IG	PS ReliefF	PS Full	Cor Relevant	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	ReliefF Highest 2	Cor Highest	ReliefF Highest	RPFS Full
	IQR	0.0814	0.0750	0.0815	0.0856	0.0820	0.0189	0.0259	0.0232	0.0251	0.0211	0.0313	0.0226	0.0278	0.0257	0.0188
Dataset 17	Median	PS Full	Spolaor	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	IG Relevant	IG Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	PS Cor	PS ReliefF	RPFS Full	PS IG
	IQR	0.0211	0.1974	0.0071	0.0070	0.0054	0.0066	0.0065	0.0073	0.0109	0.0089	0.0146	0.1887	0.1262	0.0085	0.0863
Dataset 18	Median	PS Full	PS Cor	Cor Highest	Cor Highest 2	IG Highest 2	IG Relevant	Cor Relevant	Spolaor	IG Highest	PS ReliefF	PS IG	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0278	0.0902	0.0294	0.0270	0.0301	0.0282	0.0331	0.0913	0.0368	0.0858	0.0903	0.0254	0.0276	0.0256	0.0232
Dataset 19	Median	PS Cor	Spolaor	PS ReliefF	PS IG	PS Full	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full
	IQR	0.0352	0.0326	0.0369	0.0381	0.0291	0.0051	0.0059	0.0061	0.0061	0.0070	0.0062	0.0085	0.0076	0.0121	0.0049
Dataset 20	Median	PS Cor	Spolaor	PS ReliefF	PS Full	PS IG	IG Highest	IG Relevant	IG Highest 2	Cor Highest 2	Cor Highest	Cor Relevant	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0805	0.0910	0.1162	0.0548	0.1159	0.0182	0.0155	0.0186	0.0165	0.0210	0.0135	0.0166	0.0152	0.0168	0.0155
Dataset 21	Median	Spolaor	PS ReliefF	PS Full	PS Cor	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Relevant	IG Highest	IG Highest 2	ReliefF Highest	PS IG	RPFS Full
	IQR	0.0249	0.0346	0.0200	0.2992	0.0041	0.0041	0.0041	0.0042	0.0042	0.0043	0.0041	0.0043	0.0042	0.1671	0.0085
Dataset 22	Median	Spolaor	PS Full	PS Cor	Cor Highest 2	Cor Relevant	Cor Highest	PS ReliefF	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	PS IG	RPFS Full
	IQR	0.1196	0.0735	0.2233	0.0209	0.0190	0.0258	0.2045	0.0243	0.0230	0.0289	0.0289	0.0271	0.0268	0.1456	0.0308
Dataset 23	Median	PS ReliefF	Spolaor	PS Cor	PS IG	PS Full	IG Relevant	IG Highest 2	IG Highest	ReliefF Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full
	IQR	0.0385	0.0388	0.0384	0.0406	0.0377	0.0059	0.0059	0.0058	0.0057	0.0062	0.0071	0.0090	0.0147	0.0253	0.0071
Dataset 24	Median	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full	Cor Relevant	IG Highest	Cor Highest	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full
	IQR	0.0874	0.0846	0.1297	0.1252	0.0971	0.0167	0.0215	0.0215	0.0205	0.0186	0.0217	0.0227	0.0214	0.0241	0.0183

Table 5.43: Recall.

	Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dataset 1	**Median**	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Highest	IG Highest	RPFS Full	PS Full	PS Cor	PS ReliefF	PS IG	Spolaor
	IQR	0.0213	0.0243	0.0252	0.0254	0.0296	0.0323	0.0383	0.0434	0.0329	0.0210	0.0473	0.0769	0.1224	0.0636	0.1211
Dataset 2	**Median**	Cor Relevant	IG Relevant	Cor Highest 2	IG Highest 2	ReliefF Relevant	Cor Highest	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full	PS Full	PS Cor	PS ReliefF	Spolaor	PS IG
	IQR	0.0184	0.0191	0.0204	0.0241	0.0227	0.0318	0.0365	0.0321	0.0456	0.0178	0.0040	0.0535	0.0645	0.0040	0.0052
Dataset 3	**Median**	IG Relevant	IG Highest 2	IG Highest	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	RPFS Full	Spolaor	PS Full	PS Cor	PS IG	PS ReliefF
	IQR	0.0047	0.0047	0.0056	0.0050	0.0050	0.0058	0.0071	0.0079	0.0118	0.0126	0.0656	0.0444	0.1665	0.1204	0.1453
Dataset 4	**Median**	ReliefF Relevant	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest 2	ReliefF Highest 2	Cor Highest	ReliefF Highest	PS Cor	Spolaor	RPFS Full	PS ReliefF	PS IG	PS Full
	IQR	0.0082	0.0068	0.0084	0.0067	0.0109	0.0102	0.0169	0.0264	0.0341	0.1440	0.1440	0.0159	0.1286	0.1270	0.1206
Dataset 5	**Median**	Cor Relevant	Cor Highest 2	IG Highest 2	Cor Highest	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full	PS Full	Spolaor	PS Cor	PS IG	PS ReliefF
	IQR	0.0158	0.0150	0.0180	0.0156	0.0171	0.0239	0.0204	0.0241	0.0282	0.0187	0.0584	0.0892	0.1454	0.1037	0.1143
Dataset 6	**Median**	Cor Relevant	IG Relevant	Cor Highest 2	ReliefF Relevant	IG Highest 2	ReliefF Highest 2	Cor Highest	IG Highest	ReliefF Highest	RPFS Full	Spolaor	PS Cor	PS Full	PS IG	PS ReliefF
	IQR	0.0196	0.0196	0.0251	0.0196	0.0244	0.0251	0.0396	0.0328	0.0407	0.0170	0.0823	0.0902	0.0696	0.0766	0.0817
Dataset 7	**Median**	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Relevant	Cor Highest 2	Cor Highest	RPFS Full	Spolaor	PS Full	IG Relevant	IG Highest 2	IG Highest	PS ReliefF	PS Cor	PS IG
	IQR	0.0076	0.0076	0.0085	0.0084	0.0084	0.0077	0.0204	0.0416	0.0420	0.0526	0.0522	0.0613	0.1941	0.2062	0.1962
Dataset 8	**Median**	IG Relevant	IG Highest 2	ReliefF Relevant	IG Highest	Cor Relevant	ReliefF Highest 2	Cor Highest 2	ReliefF Highest	Cor Highest	RPFS Full	Spolaor	PS Cor	PS ReliefF	PS IG	PS Full
	IQR	0.0100	0.0102	0.0095	0.0124	0.0127	0.0106	0.0179	0.0129	0.0252	0.0127	0.1165	0.1254	0.1372	0.1291	0.1112
Dataset 9	**Median**	ReliefF Relevant	ReliefF Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Highest	IG Highest	RPFS Full	PS Full	PS Cor	PS ReliefF	PS IG	Spolaor
	IQR	0.0056	0.0056	0.0040	0.0040	0.0040	0.0056	0.0062	0.0143	0.0195	0.0126	0.0237	0.0988	0.0924	0.0645	0.2357
Dataset 10	**Median**	Cor Relevant	Cor Highest 2	Cor Highest	IG Highest	IG Highest 2	IG Relevant	ReliefF Highest 2	ReliefF Highest	ReliefF Relevant	RPFS Full	PS Full	PS Cor	PS IG	Spolaor	PS ReliefF
	IQR	0.0489	0.0466	0.0573	0.0459	0.0434	0.0416	0.0439	0.0521	0.0423	0.0449	0.0556	0.0754	0.0840	0.0718	0.0700
Dataset 11	**Median**	IG Relevant	Cor Relevant	ReliefF Relevant	Cor Highest 2	Cor Highest	IG Highest 2	ReliefF Highest 2	IG Highest	ReliefF Highest	RPFS Full	PS Cor	PS ReliefF	PS IG	Spolaor	PS Full
	IQR	0.0087	0.0074	0.0088	0.0090	0.0205	0.0258	0.0349	0.0389	0.0399	0.0101	0.0436	0.0446	0.0452	0.0515	0.0403
Dataset 12	**Median**	IG Highest	IG Highest 2	IG Relevant	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	PS Cor	RPFS Full	Spolaor	PS ReliefF	PS Full	PS IG
	IQR	0.0382	0.0347	0.0404	0.0313	0.0331	0.0347	0.0490	0.0503	0.0473	0.1151	0.0174	0.1178	0.1078	0.1078	0.1193
Dataset 13	**Median**	Spolaor	ReliefF Relevant	IG Relevant	IG Highest	IG Highest 2	Cor Relevant	Cor Highest	Cor Highest 2	ReliefF Highest 2	ReliefF Highest	RPFS Full	PS ReliefF	PS Full	PS Cor	PS IG
	IQR	0.0287	0.0019	0.0016	0.0016	0.0016	0.0020	0.0020	0.0020	0.0020	0.0039	0.0065	0.0612	0.0311	0.1329	0.1264
Dataset 14	**Median**	Cor Highest 2	Cor Highest	Cor Relevant	IG Highest 2	IG Relevant	IG Highest	ReliefF Highest 2	ReliefF Relevant	ReliefF Highest	RPFS Full	PS Full	Spolaor	PS Cor	PS ReliefF	PS IG
	IQR	0.0402	0.0456	0.0394	0.0496	0.0477	0.0469	0.0613	0.0478	0.0712	0.0550	0.0651	0.1327	0.0884	0.1251	0.1039
Dataset 15	**Median**	PS Cor	PS ReliefF	Spolaor	IG Relevant	ReliefF Relevant	Cor Relevant	Cor Highest 2	IG Highest 2	ReliefF Highest 2	PS IG	Cor Highest	IG Highest	ReliefF Highest	RPFS Full	PS Full
	IQR	0.0739	0.0706	0.0725	0.0035	0.0038	0.0038	0.0043	0.0037	0.0046	0.0641	0.0080	0.0068	0.0075	0.0103	0.0568
Dataset 16	**Median**	Cor Relevant	Cor Highest 2	IG Highest 2	IG Relevant	IG Highest	ReliefF Relevant	ReliefF Highest 2	ReliefF Highest	Cor Highest	Spolaor	PS Cor	PS IG	PS ReliefF	RPFS Full	PS Full
	IQR	0.0360	0.0469	0.0550	0.0499	0.0554	0.0439	0.0496	0.0515	0.0582	0.1240	0.1253	0.1512	0.1148	0.0320	0.0985
Dataset 17	**Median**	Cor Relevant	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	Cor Highest 2	IG Highest	ReliefF Highest	Cor Highest	RPFS Full	PS Full	Spolaor	PS ReliefF	PS Cor	PS IG
	IQR	0.0069	0.0077	0.0078	0.0106	0.0105	0.0092	0.0146	0.0138	0.0201	0.0128	0.0260	0.1686	0.1209	0.1438	0.0933
Dataset 18	**Median**	Cor Highest 2	Cor Highest	IG Highest 2	IG Relevant	Cor Relevant	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full	PS Full	PS Cor	Spolaor	PS IG	PS ReliefF
	IQR	0.0479	0.0558	0.0596	0.0546	0.0602	0.0666	0.0517	0.0465	0.0469	0.0380	0.0804	0.1083	0.0948	0.1008	0.1031
Dataset 19	**Median**	IG Relevant	IG Highest 2	Cor Relevant	Cor Highest 2	Cor Highest	ReliefF Relevant	IG Highest	ReliefF Highest 2	ReliefF Highest	RPFS Full	PS Cor	Spolaor	PS IG	PS ReliefF	PS Full
	IQR	0.0082	0.0086	0.0095	0.0095	0.0113	0.0106	0.0153	0.0112	0.0188	0.0124	0.0555	0.0568	0.0575	0.0681	0.0447
Dataset 20	**Median**	IG Highest	IG Relevant	IG Highest 2	Cor Highest 2	Cor Relevant	Cor Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full	Spolaor	PS Cor	PS Full	PS ReliefF	PS IG
	IQR	0.0394	0.0300	0.0396	0.0343	0.0262	0.0419	0.0360	0.0311	0.0329	0.0315	0.1113	0.1370	0.1163	0.1231	0.1556
Dataset 21	**Median**	Cor Relevant	Cor Highest	Cor Highest 2	IG Relevant	IG Highest 2	ReliefF Relevant	ReliefF Highest 2	IG Highest	ReliefF Highest	Spolaor	PS ReliefF	RPFS Full	PS Full	PS Cor	PS IG
	IQR	0.0019	0.0019	0.0019	0.0019	0.0019	0.0023	0.0023	0.0023	0.0020	0.0343	0.0519	0.0109	0.0285	0.2221	0.1590
Dataset 22	**Median**	Cor Highest 2	Cor Relevant	Cor Highest	IG Relevant	IG Highest 2	IG Highest	ReliefF Highest	ReliefF Highest 2	ReliefF Relevant	RPFS Full	Spolaor	PS Full	PS Cor	PS ReliefF	PS IG
	IQR	0.0397	0.0400	0.0440	0.0410	0.0405	0.0438	0.0489	0.0496	0.0430	0.0607	0.1078	0.0726	0.1529	0.1397	0.1486
Dataset 23	**Median**	ReliefF Relevant	Cor Relevant	IG Relevant	IG Highest 2	IG Highest	Cor Highest 2	PS Cor	PS ReliefF	Spolaor	Cor Highest	ReliefF Highest 2	PS IG	ReliefF Highest	RPFS Full	PS Full
	IQR	0.0038	0.0031	0.0038	0.0038	0.0040	0.0055	0.0794	0.0800	0.0800	0.0103	0.0250	0.0779	0.0452	0.0102	0.0849
Dataset 24	**Median**	IG Relevant	IG Highest 2	Cor Relevant	IG Highest	Cor Highest	Cor Highest 2	ReliefF Highest	ReliefF Relevant	ReliefF Highest 2	RPFS Full	Spolaor	PS Cor	PS IG	PS ReliefF	PS Full
	IQR	0.0352	0.0381	0.0301	0.0371	0.0384	0.0331	0.0454	0.0495	0.0447	0.0439	0.1784	0.1711	0.1851	0.1545	0.1295

From the above tables, it can be seen that the results observed for Dataset 1 are not typical. Each of the tables will be discussed in detail and the deviations from the results observed in Dataset 1 will be noted.

For Hamming-loss in Table 5.40, the median values of PS ReliefF, PS Correlation, and Spolaôr are smaller than those of the RPFS procedures, with the exception of Datasets 1 and 14. Not all of the PS techniques perform well, as the procedure PS IG is ranked last for eight of the 24 datasets. The IQRs for PS and Spolaôr are typically larger, with the exception of Datasets 11 and 16, where there is more variation in the results of the RPFS procedures. The median of the PS full model is smaller than those of the reduced PS models for eleven datasets and the Spolaôr procedure for eight of the datasets. The medians of the reduced RPFS models are smaller than that of the corresponding full model for all relevance measures and datasets except for Dataset 7, where RPFS IG has a larger median than RPFS Full. It is interesting to note that the four lowest ranked procedures are all based on IG for Dataset 7.

From Table 5.41 for One-error, the medians of the RPFS procedures are smaller than those of PS and Spolaôr, with the exception of Datasets 11 and 19 where PS Correlation ranks highest. These results are in line with the results observed for Dataset 1. The IQRs of the RPFS procedures are larger for Datasets 2, 4, 8, 10, 11, 15, 18, and 23. The median for PS Full is smaller than the medians of the reduced PS procedures where a noise ratio of 1:1 is present, but as soon as the noise ratio increases to 5:1, the rankings of the PS procedures improve. This will be investigated further in Table 5.46 below.

Considering the values for Precision in Table 5.42, the medians of the RPFS procedures are larger than the median Precision values of PS and Spolaôr for Dataset 1, 2, 9, 10, and 14. The PS and Spolaôr procedures perform better on the other datasets. The Spolaôr method tends to rank higher than the PS procedures. The IQRs of the PS and Spolaôr procedures are consistently larger than the IQRs of the RPFS procedures. The median of the PS full model is larger than those of the reduced PS models for ten datasets. The median of PS procedure for the full set of features is also larger than the median of the Spolaôr procedure for eight of the datasets. The medians of the reduced RPFS models are larger than that of the full model for all importance measures and datasets except for Dataset 7, where RPFS IG has a smaller median than RPFS Full.

For Recall in Table 5.43, the median values of PS and Spolaôr are smaller than those of the RPFS procedures, with the exception of Dataset 15. One also consistently observes more variation in the results of PS and Spolaôr across all 24 datasets. This is the same conclusion that can be drawn based on Dataset 1. The majority of the FS procedures based on RPFS perform better than the full model, while the reduced PS models do not consistently outperform the full PS model.

In order to gain better insight into these results, the Method of Pairwise Comparisons is used to obtain a ranking of the procedures for all 24 datasets. The results are summarised in Table 5.44. The median IQR values are used to provide an indication of the variation. Again, the darker the green, the larger the median IQR (and the variation).

As in Section 5.4, the procedures are also assessed in terms of the effect of a change in one of the properties associated with multi-label datasets. The datasets are therefore grouped according to the signal strength (Table 5.45), the number of irrelevant features (Table 5.46), the number of training instances (Tables 5.47 and 5.48), the label dependence (Table 5.49), and the vector of label densities (Table 5.50). The medians are then ranked using the Method of Pairwise Comparisons.

The same pattern in the rankings is observed in Tables 5.44 to 5.48 as in Tables 5.40 to 5.43: the RPFS procedures rank higher for One-error and Recall over any grouping of the datasets, while PS and Spolaôr rank higher for Hamming-loss and Precision (with the exception of PS IG for Precision). The RPFS procedures are more stable (*i.e.*, show less variation) than PS and Spolaôr for all groupings of the datasets.

Some interesting observations can be made from these tables. For example, in Table 5.45, for the datasets grouped based on the strength of the signal, the ranking of Spolaôr improves with an increase in the signal strength from 10 to 100 for all four evaluation measures. For Precision, the procedure based on PS ReliefF improves substantially if the number of irrelevant features increases. PS ReliefF ranks ninth for the cases where there are only ten irrelevant features. This ranking improves to third when the number of irrelevant features increases to 50. An increase in the number of irrelevant features implies that the number of training instances also increases.

Considering the results for grouping based on label dependence summarised in Table 5.49, the RPFS procedures rank highest for One-error and Precision, as before. For Hamming-loss and

Recall, the rankings need to be considered more carefully. For the cases where no label correlation is present, the methods based on RPFS, especially those that utilise the correlation coefficient as relevance measures, rank fairly high. However, as the correlation between labels is increased to 0.4, the rankings of methods based on PS improve significantly.

In Table 5.50, the results for groupings based on different vectors of densities are presented. For One-error and Recall, the RPFS procedures rank higher than those based on PS and Spolaôr. However, for Hamming-loss, PS and Spolaôr perform better. For Precision, the rankings of methods based on PS depend on the vector of densities. The rankings change significantly depending on the densities selected.

Table 5.44: Method of Pairwise Comparisons for all 24 datasets.

Rank	Hamming Loss		One-Error		Precision		Recall	
	Technique	Median IQR	Technique	Median IQR	Technique	Median IQR	Technique	Median IQR
1	Spolaor	0.01769	Cor Relevant	0.01859	Spolaor	0.06538	Cor Relevant	0.01426
2	PS Full	0.01251	IG Relevant	0.02124	PS Full	0.03739	IG Relevant	0.01810
3	PS Cor	0.04897	Cor Highest 2	0.02056	PS Cor	0.08605	Cor Highest 2	0.01647
4	PS ReliefF	0.03723	IG Highest 2	0.02925	PS ReliefF	0.08574	IG Highest 2	0.02423
5	Cor Relevant	0.00738	Cor Highest	0.03240	Cor Relevant	0.00845	ReliefF Relevant	0.01509
6	PS IG	0.05836	ReliefF Relevant	0.01799	IG Relevant	0.00908	Cor Highest	0.02578
7	IG Relevant	0.00937	IG Highest	0.03649	Cor Highest 2	0.01013	IG Highest	0.03243
8	Cor Highest 2	0.00954	ReliefF Highest 2	0.02845	IG Highest 2	0.01252	ReliefF Highest 2	0.02506
9	IG Highest 2	0.01198	ReliefF Highest	0.03587	Cor Highest	0.01547	ReliefF Highest	0.03711
10	ReliefF Relevant	0.00795	Spolaor	0.03230	ReliefF Relevant	0.00870	RPFS Full	0.01719
11	Cor Highest	0.01453	PS Full	0.01851	PS IG	0.08832	Spolaor	0.09201
12	IG Highest	0.01565	RPFS Full	0.02191	ReliefF Highest 2	0.01425	PS Full	0.05760
13	ReliefF Highest 2	0.01474	PS Cor	0.07789	IG Highest	0.01778	PS Cor	0.12017
14	ReliefF Highest	0.01890	PS IG	0.07236	ReliefF Highest	0.01836	PS ReliefF	0.11109
15	RPFS Full	0.00916	PS ReliefF	0.06396	RPFS Full	0.01070	PS IG	0.10381

Table 5.45: Method of Pairwise Comparisons for the signal level.

Rank	Hamming Loss				One-Error				Precision				Recall			
	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR	Signal 10	Median IQR	Signal 100	Median IQR
1	PS Full	0.01150	Spolaor	0.01637	Cor Relevant	0.02116	Cor Relevant	0.00921	PS Full	0.02926	Spolaor	0.05287	Cor Relevant	0.01899	Cor Relevant	0.00750
2	PS Cor	0.04897	PS Cor	0.05901	IG Relevant	0.02219	IG Relevant	0.01388	Spolaor	0.08952	PS Cor	0.12419	IG Relevant	0.01932	IG Relevant	0.00842
3	Spolaor	0.03869	PS ReliefF	0.02531	Cor Highest 2	0.02650	ReliefF Relevant	0.00895	PS Cor	0.08399	PS ReliefF	0.07039	Cor Highest 2	0.02276	IG Highest 2	0.00930
4	PS ReliefF	0.04592	PS Full	0.01361	IG Highest 2	0.02996	Cor Highest 2	0.01249	Cor Relevant	0.00999	PS Full	0.04200	IG Highest 2	0.02491	ReliefF Relevant	0.00785
5	Cor Relevant	0.00915	Cor Relevant	0.00531	Cor Highest	0.03970	IG Highest 2	0.01516	IG Relevant	0.01075	Cor Relevant	0.00688	Cor Highest	0.03326	Cor Highest 2	0.00926
6	IG Relevant	0.01029	PS IG	0.05976	ReliefF Highest 2	0.02993	IG Highest	0.01926	Cor Highest 2	0.01036	Cor Highest 2	0.00822	ReliefF Relevant	0.02342	IG Highest	0.01168
7	Cor Highest 2	0.01059	IG Relevant	0.00561	ReliefF Relevant	0.02640	Cor Highest	0.01986	IG Highest 2	0.01488	PS IG	0.11216	IG Highest	0.03286	Cor Highest	0.01776
8	PS IG	0.05836	Cor Highest 2	0.00641	IG Highest	0.03745	ReliefF Highest 2	0.02201	Cor Highest	0.01668	IG Relevant	0.00775	ReliefF Highest 2	0.03173	ReliefF Highest 2	0.01375
9	IG Highest 2	0.01364	IG Highest 2	0.00582	ReliefF Highest	0.03587	Spolaor	0.01211	ReliefF Relevant	0.01394	IG Highest 2	0.00832	ReliefF Highest	0.03911	ReliefF Highest	0.02352
10	Cor Highest	0.01643	ReliefF Relevant	0.00558	PS Full	0.02360	ReliefF Highest	0.02813	IG Highest	0.01802	ReliefF Relevant	0.00630	RPFS Full	0.01756	Spolaor	0.09391
11	ReliefF Relevant	0.01315	IG Highest	0.00775	RPFS Full	0.02405	PS Cor	0.10530	ReliefF Highest 2	0.01647	Cor Highest	0.01153	PS Full	0.05144	RPFS Full	0.01428
12	ReliefF Highest 2	0.01531	Cor Highest	0.01019	PS Cor	0.05544	PS ReliefF	0.05995	PS ReliefF	0.09255	IG Highest	0.01059	PS Cor	0.09449	PS Cor	0.13843
13	IG Highest	0.01714	ReliefF Highest 2	0.00849	Spolaor	0.04838	PS Full	0.01596	PS IG	0.08548	ReliefF Highest 2	0.01030	Spolaor	0.09201	PS ReliefF	0.12686
14	ReliefF Highest	0.01890	ReliefF Highest	0.01561	PS IG	0.05411	PS IG	0.12214	ReliefF Highest	0.01979	ReliefF Highest	0.01417	PS ReliefF	0.09775	PS Full	0.06885
15	RPFS Full	0.00921	RPFS Full	0.00876	PS ReliefF	0.06396	RPFS Full	0.01801	RPFS Full	0.01108	RPFS Full	0.00945	PS IG	0.08029	PS IG	0.12802

Table 5.46: Method of Pairwise Comparisons for the number of irrelevant features.

	Hamming Loss				One-Error				Precision				Recall			
Rank	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR	Irrelevant features = 10	Median IQR	Irrelevant features = 50	Median IQR
1	Spolaor	0.01637	Spolaor	0.02108	Cor Relevant	0.01859	Cor Relevant	0.02032	Spolaor	0.05287	Spolaor	0.08442	Cor Relevant	0.01426	Cor Relevant	0.01782
2	PS Full	0.01251	PS Cor	0.03628	IG Relevant	0.02124	IG Relevant	0.02474	PS Full	0.03933	PS Cor	0.08883	IG Relevant	0.01810	IG Relevant	0.01937
3	PS Cor	0.06170	PS ReliefF	0.03290	Cor Highest 2	0.02056	Cor Highest 2	0.02500	Cor Relevant	0.00845	PS ReliefF	0.08574	Cor Highest 2	0.01647	Cor Highest 2	0.02127
4	PS ReliefF	0.05224	PS Full	0.01129	IG Highest 2	0.02553	IG Highest 2	0.03210	PS Cor	0.06481	PS Full	0.03395	IG Highest 2	0.02102	IG Highest 2	0.03025
5	Cor Relevant	0.00738	PS IG	0.05929	ReliefF Relevant	0.01779	Cor Highest	0.03660	IG Relevant	0.00908	Cor Relevant	0.00982	ReliefF Relevant	0.01455	Cor Highest	0.02764
6	IG Relevant	0.00937	Cor Relevant	0.00862	ReliefF Highest 2	0.02670	IG Highest	0.03745	Cor Highest 2	0.01013	Cor Highest 2	0.01182	ReliefF Highest 2	0.01866	IG Highest	0.03766
7	Cor Highest 2	0.00954	IG Relevant	0.01026	Cor Highest	0.03090	ReliefF Relevant	0.02151	IG Highest 2	0.01149	IG Relevant	0.01111	Cor Highest	0.02578	ReliefF Relevant	0.02178
8	IG Highest 2	0.01038	Cor Highest 2	0.01046	IG Highest	0.03141	ReliefF Highest 2	0.03596	ReliefF Relevant	0.00858	PS IG	0.08832	IG Highest	0.02799	ReliefF Highest 2	0.03302
9	ReliefF Relevant	0.00795	IG Highest 2	0.01575	ReliefF Highest	0.03400	ReliefF Highest	0.04184	PS ReliefF	0.10203	Cor Highest	0.01767	ReliefF Highest	0.03112	ReliefF Highest	0.04259
10	ReliefF Highest 2	0.01174	Cor Highest	0.01474	Spolaor	0.01453	Spolaor	0.04838	ReliefF Highest 2	0.01248	IG Highest 2	0.01762	RPFS Full	0.01738	RPFS Full	0.01511
11	Cor Highest	0.01453	IG Highest	0.01759	PS Full	0.01818	PS Full	0.01881	Cor Highest	0.01521	IG Highest	0.01924	Spolaor	0.08577	Spolaor	0.10132
12	IG Highest	0.01445	ReliefF Relevant	0.01082	RPFS Full	0.02191	PS Cor	0.07789	IG Highest	0.01590	ReliefF Relevant	0.01195	PS Full	0.05284	PS Cor	0.11166
13	ReliefF Highest	0.01739	ReliefF Highest 2	0.01593	PS Cor	0.06643	PS IG	0.07440	ReliefF Highest	0.01784	ReliefF Highest 2	0.01647	PS Cor	0.13468	PS ReliefF	0.09775
14	PS IG	0.05430	ReliefF Highest	0.02137	PS IG	0.05658	PS ReliefF	0.05269	PS IG	0.09147	ReliefF Highest	0.02260	PS ReliefF	0.12554	PS Full	0.06094
15	RPFS Full	0.00913	RPFS Full	0.00921	PS ReliefF	0.07806	RPFS Full	0.02065	RPFS Full	0.01070	RPFS Full	0.00982	PS IG	0.11207	PS IG	0.10234

Table 5.47: Method of Pairwise Comparisons for the number of training instances: Hamming-loss and One-error.

	Hamming Loss						One-Error					
Rank	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR
1	Spolaor	0.03869	Spolaor	0.01637	Spolaor	0.00967	Cor Relevant	0.05364	Cor Relevant	0.01859	IG Relevant	0.00683
2	PS Cor	0.03628	PS Full	0.01251	PS Cor	0.05351	Cor Highest 2	0.06043	IG Relevant	0.02124	ReliefF Relevant	0.00696
3	PS Full	0.01817	PS Cor	0.06170	PS ReliefF	0.01507	IG Highest 2	0.05281	Cor Highest 2	0.02056	Cor Relevant	0.00706
4	PS ReliefF	0.04592	PS ReliefF	0.05224	PS Full	0.00768	IG Relevant	0.05310	IG Highest 2	0.02553	IG Highest 2	0.00718
5	PS IG	0.06536	Cor Relevant	0.00738	IG Relevant	0.00303	Cor Highest	0.06688	ReliefF Relevant	0.01779	Cor Highest 2	0.00847
6	Cor Highest 2	0.02339	IG Relevant	0.00937	Cor Relevant	0.00317	IG Highest	0.06211	ReliefF Highest 2	0.02670	Cor Highest	0.01243
7	Cor Relevant	0.01959	Cor Highest 2	0.00954	ReliefF Relevant	0.00350	ReliefF Highest 2	0.05566	Cor Highest	0.03090	ReliefF Highest 2	0.00835
8	IG Highest 2	0.02349	IG Highest 2	0.01038	Cor Highest 2	0.00358	ReliefF Relevant	0.05644	IG Highest	0.03141	IG Highest	0.01324
9	IG Relevant	0.02221	ReliefF Relevant	0.00795	PS IG	0.02983	ReliefF Highest	0.06674	ReliefF Highest	0.03400	Spolaor	0.01831
10	Cor Highest	0.02741	ReliefF Highest 2	0.01174	IG Highest 2	0.00342	Spolaor	0.06203	Spolaor	0.01453	ReliefF Highest	0.01830
11	IG Highest	0.02946	Cor Highest	0.01453	ReliefF Highest 2	0.00532	PS Full	0.05340	PS Full	0.01818	PS Cor	0.06755
12	ReliefF Highest 2	0.02640	IG Highest	0.01445	Cor Highest	0.00504	PS Cor	0.07789	RPFS Full	0.02191	PS ReliefF	0.01860
13	ReliefF Highest	0.02689	ReliefF Highest	0.01739	IG Highest	0.00508	RPFS Full	0.05446	PS Cor	0.06643	PS Full	0.01275
14	ReliefF Relevant	0.02518	PS IG	0.05430	ReliefF Highest	0.00930	PS IG	0.07440	PS IG	0.05658	PS IG	0.07153
15	RPFS Full	0.02008	RPFS Full	0.00913	RPFS Full	0.00543	PS ReliefF	0.08321	PS ReliefF	0.07806	RPFS Full	0.01355

Table 5.48: Method of Pairwise Comparisons for the number of training instances: Precision and Recall.

	Precision						Recall					
Rank	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR	Number of training instances = 30	Median IQR	Number of training instances = 80	Median IQR	Number of training instances = 240	Median IQR
1	Spolaor	0.08952	Spolaor	0.05287	Spolaor	0.03182	Cor Highest 2	0.03994	Cor Relevant	0.01426	IG Relevant	0.00469
2	PS Cor	0.08883	PS Full	0.03933	PS ReliefF	0.03635	Cor Relevant	0.03767	IG Relevant	0.01810	Cor Relevant	0.00389
3	PS Full	0.05792	Cor Relevant	0.00845	PS Cor	0.10698	IG Highest 2	0.04196	Cor Highest 2	0.01647	ReliefF Relevant	0.00472
4	PS ReliefF	0.10772	PS Cor	0.06481	PS Full	0.02508	IG Relevant	0.04130	IG Highest 2	0.02102	Cor Highest 2	0.00490
5	Cor Highest 2	0.02339	IG Relevant	0.00908	ReliefF Relevant	0.00575	Cor Highest	0.04482	ReliefF Relevant	0.01455	IG Highest 2	0.00500
6	Cor Relevant	0.01895	Cor Highest 2	0.01013	Cor Relevant	0.00515	IG Highest	0.04485	ReliefF Highest 2	0.01866	Cor Highest	0.00917
7	Cor Highest	0.02679	IG Highest 2	0.01149	IG Relevant	0.00552	ReliefF Highest	0.05017	Cor Highest	0.02578	ReliefF Highest 2	0.00804
8	PS IG	0.10144	ReliefF Relevant	0.00858	PS IG	0.06261	ReliefF Highest 2	0.04805	IG Highest	0.02799	IG Highest	0.01072
9	IG Highest	0.03008	PS ReliefF	0.10203	Cor Highest 2	0.00592	ReliefF Relevant	0.04541	ReliefF Highest	0.03112	ReliefF Highest	0.01405
10	IG Highest 2	0.02473	ReliefF Highest 2	0.01248	ReliefF Highest 2	0.00640	RPFS Full	0.04094	RPFS Full	0.01738	RPFS Full	0.01059
11	IG Relevant	0.02313	Cor Highest	0.01521	IG Highest 2	0.00594	Spolaor	0.11451	Spolaor	0.08577	Spolaor	0.06463
12	ReliefF Highest 2	0.02656	IG Highest	0.01590	Cor Highest	0.00690	PS Full	0.08949	PS Full	0.05284	PS ReliefF	0.06933
13	ReliefF Highest	0.02617	ReliefF Highest	0.01784	IG Highest	0.00715	PS Cor	0.12017	PS Cor	0.13468	PS Cor	0.08912
14	ReliefF Relevant	0.02520	PS IG	0.09147	ReliefF Highest	0.00997	PS IG	0.13391	PS ReliefF	0.12554	PS Full	0.03567
15	RPFS Full	0.02100	RPFS Full	0.01070	RPFS Full	0.00778	PS ReliefF	0.11897	PS IG	0.11207	PS IG	0.07121

Table 5.49: Method of Pairwise Comparisons for the label dependence.

	Hamming Loss				One-Error				Precision				Recall			
Rank	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR	ρ = 0	Median IQR	ρ = 0.4	Median IQR
1	PS Full	0.01246	Spolaor	0.01548	Cor Relevant	0.01583	IG Relevant	0.01785	PS Full	0.02930	Spolaor	0.04994	Cor Relevant	0.01207	IG Relevant	0.01456
2	Spolaor	0.05717	PS Cor	0.01493	Cor Highest 2	0.01643	Cor Relevant	0.01859	Spolaor	0.10543	PS Cor	0.05082	Cor Highest 2	0.01213	Cor Relevant	0.01558
3	Cor Relevant	0.00765	PS ReliefF	0.01695	Cor Highest	0.02780	IG Highest 2	0.02925	Cor Relevant	0.00832	PS ReliefF	0.05320	Cor Highest	0.01780	IG Highest 2	0.02423
4	Cor Highest 2	0.00765	PS IG	0.01636	IG Highest 2	0.02589	Cor Highest 2	0.02056	Cor Highest 2	0.00879	PS IG	0.05180	IG Relevant	0.02071	Cor Highest 2	0.01916
5	Cor Highest	0.01105	PS Full	0.01251	IG Relevant	0.02463	ReliefF Relevant	0.01799	Cor Highest	0.01245	PS Full	0.04961	IG Highest 2	0.02170	ReliefF Relevant	0.01509
6	PS Cor	0.09874	IG Relevant	0.00731	IG Highest	0.03640	IG Highest	0.03649	IG Relevant	0.01560	IG Relevant	0.00843	IG Highest	0.02842	IG Highest	0.03243
7	IG Highest 2	0.01461	IG Highest 2	0.01198	ReliefF Relevant	0.01918	Cor Highest	0.03626	IG Highest 2	0.01497	Cor Relevant	0.00845	ReliefF Relevant	0.01735	Cor Highest	0.02910
8	IG Relevant	0.01457	Cor Relevant	0.00738	ReliefF Highest 2	0.01698	Spolaor	0.02873	IG Highest	0.01959	Cor Highest 2	0.01015	ReliefF Highest 2	0.01545	ReliefF Highest 2	0.02810
9	IG Highest	0.01899	Cor Highest 2	0.00957	ReliefF Highest	0.02784	ReliefF Highest 2	0.03065	PS Cor	0.17592	IG Highest 2	0.01252	ReliefF Highest	0.02123	ReliefF Highest	0.04032
10	ReliefF Relevant	0.01088	ReliefF Relevant	0.00750	RPFS Full	0.01711	PS Cor	0.02045	ReliefF Relevant	0.01147	ReliefF Relevant	0.00858	RPFS Full	0.01952	RPFS Full	0.01645
11	ReliefF Highest 2	0.01053	IG Highest	0.01565	PS Full	0.02299	ReliefF Highest	0.03675	ReliefF Highest 2	0.01040	Cor Highest	0.01668	PS Full	0.04581	PS Cor	0.10261
12	PS ReliefF	0.07059	Cor Highest	0.01600	Spolaor	0.08024	PS ReliefF	0.03175	ReliefF Highest	0.01456	IG Highest	0.01778	Spolaor	0.09201	Spolaor	0.09679
13	ReliefF Highest	0.01417	ReliefF Highest 2	0.01531	PS IG	0.14699	PS IG	0.02625	PS ReliefF	0.12849	ReliefF Highest 2	0.01495	PS Cor	0.13832	PS ReliefF	0.09477
14	RPFS Full	0.01193	ReliefF Highest	0.02063	PS Cor	0.13416	PS Full	0.01818	RPFS Full	0.01252	ReliefF Highest	0.02009	PS ReliefF	0.11764	PS IG	0.09860
15	PS IG	0.07551	RPFS Full	0.00795	PS ReliefF	0.11471	RPFS Full	0.02316	PS IG	0.10992	RPFS Full	0.00861	PS IG	0.10381	PS Full	0.09171

Table 5.50: Method of Pairwise Comparisons for different vectors of densities.

	Hamming Loss				One-Error				Precision				Recall			
Rank	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR	Fixed	Median IQR	Varied	Median IQR
1	Spolaor	0.01769	Spolaor	0.02037	Cor Relevant	0.01859	IG Relevant	0.02124	Spolaor	0.06497	Spolaor	0.07035	Cor Relevant	0.01288	Cor Relevant	0.01426
2	PS Full	0.01213	PS Cor	0.03628	IG Relevant	0.01869	Cor Relevant	0.01598	PS Full	0.03061	PS Cor	0.08737	IG Relevant	0.01389	IG Relevant	0.01834
3	PS Cor	0.05593	PS Full	0.01310	Cor Highest 2	0.02056	Cor Highest 2	0.01848	Cor Relevant	0.00878	PS Full	0.03933	Cor Highest 2	0.01530	Cor Highest 2	0.01647
4	PS ReliefF	0.03723	PS ReliefF	0.03637	IG Highest 2	0.02925	IG Highest 2	0.02614	Cor Highest 2	0.00983	PS ReliefF	0.10101	IG Highest 2	0.02473	IG Highest 2	0.02120
5	Cor Relevant	0.00666	PS IG	0.05887	Cor Highest	0.03626	Cor Highest	0.02780	IG Relevant	0.00786	PS IG	0.11676	ReliefF Relevant	0.01577	Cor Highest	0.02261
6	IG Relevant	0.00767	IG Relevant	0.00949	ReliefF Relevant	0.01779	ReliefF Relevant	0.01849	IG Highest 2	0.01488	Cor Relevant	0.00822	Cor Highest	0.02910	ReliefF Relevant	0.01509
7	Cor Highest 2	0.00830	Cor Relevant	0.00754	ReliefF Highest 2	0.03315	IG Highest	0.03141	PS Cor	0.08235	Cor Highest 2	0.01013	IG Highest	0.03250	IG Highest	0.02834
8	IG Highest 2	0.01364	IG Highest 2	0.01038	IG Highest	0.03745	ReliefF Highest 2	0.02625	PS ReliefF	0.06845	IG Relevant	0.01075	ReliefF Highest 2	0.03363	ReliefF Highest 2	0.02272
9	ReliefF Relevant	0.00835	Cor Highest 2	0.00964	ReliefF Highest	0.03956	ReliefF Highest	0.03033	ReliefF Relevant	0.00912	Cor Highest	0.01430	ReliefF Highest	0.03911	ReliefF Highest	0.03206
10	Cor Highest	0.01600	Cor Highest	0.01326	PS Full	0.01778	Spolaor	0.03169	ReliefF Highest 2	0.01795	IG Highest 2	0.01149	RPFS Full	0.01663	RPFS Full	0.01785
11	ReliefF Highest 2	0.01634	IG Highest	0.01445	Spolaor	0.04264	PS Full	0.02194	Cor Highest	0.01681	IG Highest	0.01597	Spolaor	0.09512	Spolaor	0.09201
12	IG Highest	0.01828	ReliefF Relevant	0.00795	RPFS Full	0.02101	PS Cor	0.07320	IG Highest	0.01900	ReliefF Relevant	0.00870	PS Full	0.05144	PS Full	0.07113
13	PS IG	0.05472	ReliefF Highest 2	0.01378	PS Cor	0.07789	RPFS Full	0.02280	ReliefF Highest	0.02201	ReliefF Highest 2	0.01340	PS Cor	0.09362	PS Cor	0.14037
14	ReliefF Highest	0.02217	ReliefF Highest	0.01655	PS IG	0.06691	PS IG	0.07440	PS IG	0.07829	ReliefF Highest	0.01710	PS ReliefF	0.10010	PS ReliefF	0.11764
15	RPFS Full	0.00916	RPFS Full	0.00948	PS ReliefF	0.03856	PS ReliefF	0.06789	RPFS Full	0.01070	RPFS Full	0.01011	PS IG	0.09396	PS IG	0.11639

In the next subsection, the number of features selected by each of the FS procedures is investigated.

Feature Reduction

As before, a comparison of the Feature Reduction for each of the FS methods is included. In Figures 5.41 to 5.43, the Feature Reduction of the FS procedures are compared for each of the 24 synthetic datasets.

For Datasets 1 to 8, where there are ten relevant and ten irrelevant features included in each of the datasets, one would expect lower values of Feature Reduction when compared with the Datasets 9 – 24. The results presented in Figure 5.41, with the exception of Dataset 4 and 8, and to a lesser degree, Dataset 6, do not confirm this for the FS procedures based on PS. The FS procedures based on PS and to a lesser extent, the method proposed by Spolaôr, include fewer features than the RPFS procedures. The procedures based on Spolaôr and PS are typically associated with Feature Reduction values larger than 80.

Datasets 9 to 24 each include ten relevant and 50 irrelevant features. One would expect the FS techniques to produce higher values of Feature Reduction due to the higher proportion of noise included. This is confirmed by Figures 5.42 and 5.43. Once again, the procedures based on PS and Spolaôr include fewer features than the RPFS procedures for all datasets, except for Dataset 23.

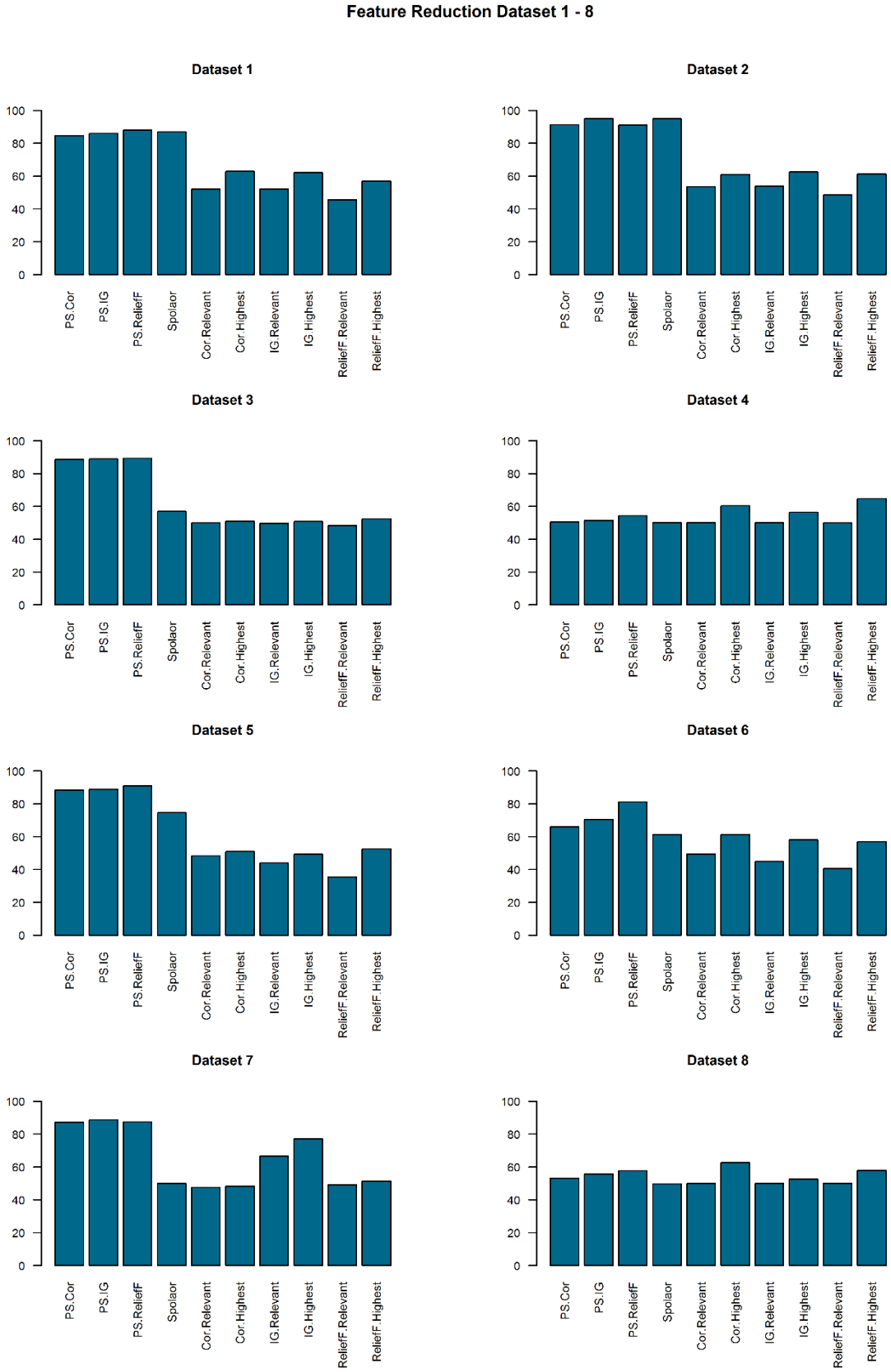


Figure 5.41 Comparison of feature reduction achieved for classification: Dataset 1 – 8.

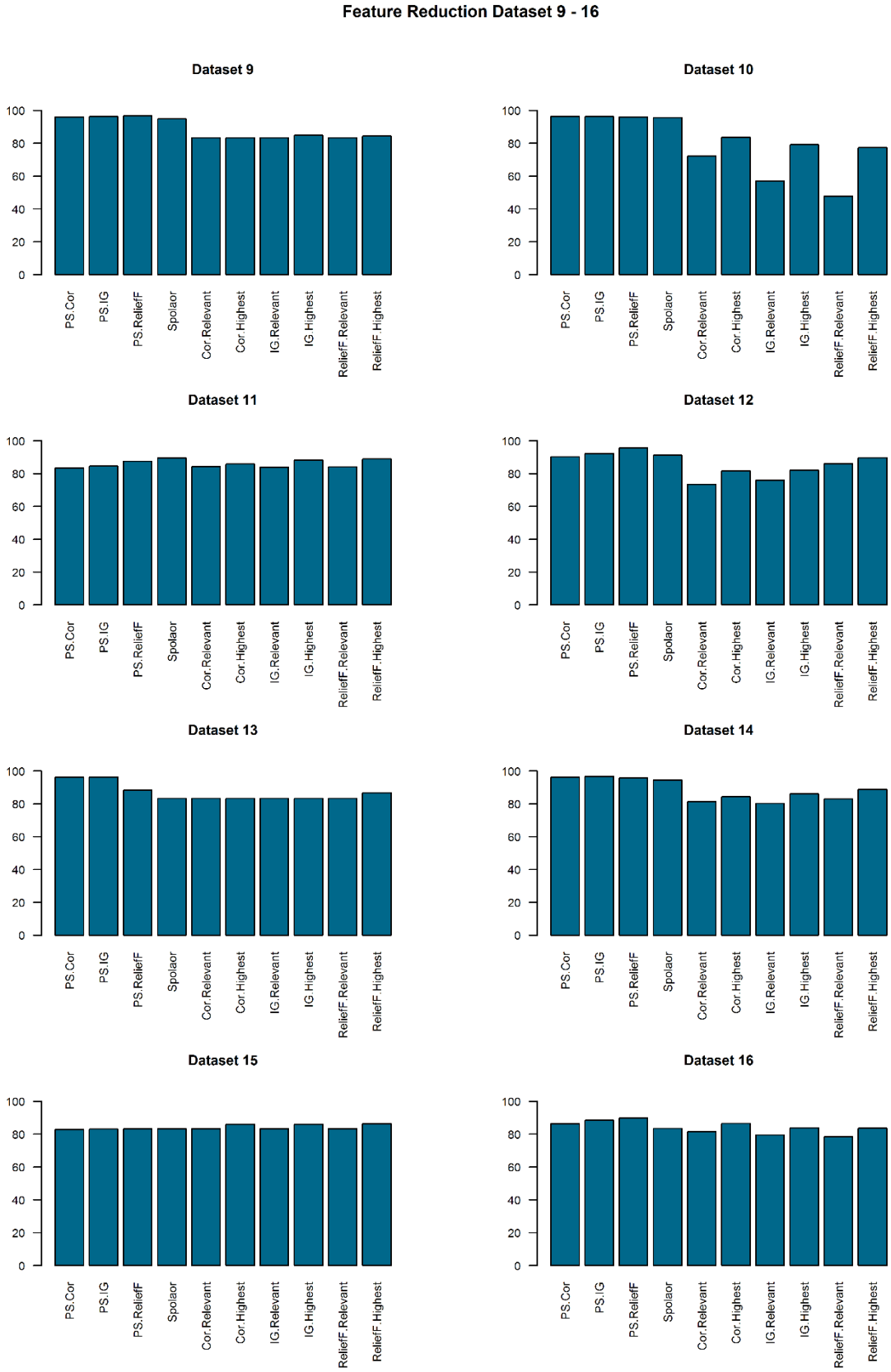


Figure 5.42 Comparison of feature reduction achieved for classification: Dataset 9 – 16.

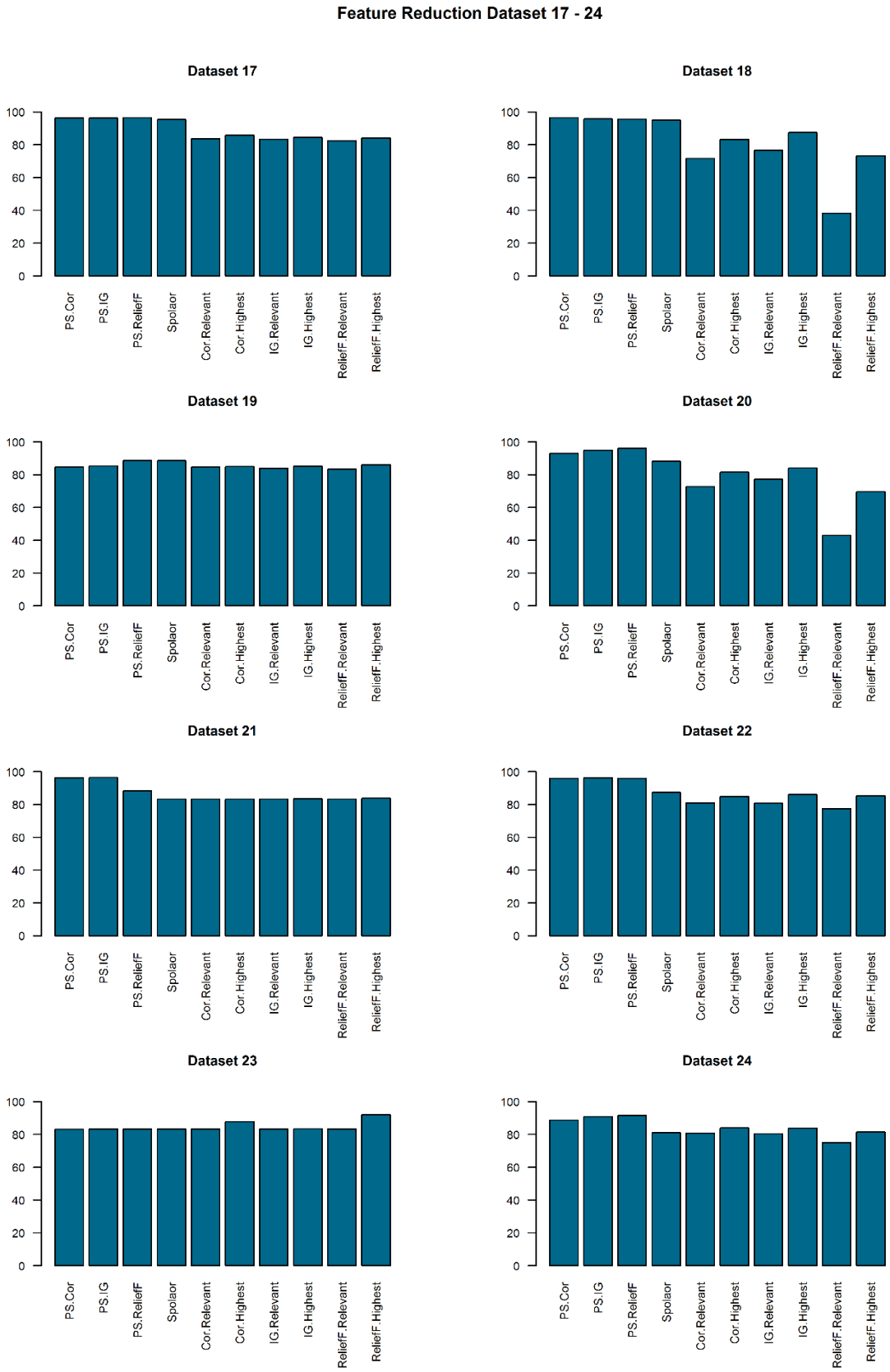


Figure 5.43 Comparison of feature reduction achieved for classification: Dataset 17 – 24.

Some important observations regarding the results in this section are:

1) The proposed RPFS procedure performs better than the two established techniques based on PS and the proposal by Spolaôr for One-error and for Recall. However, for Hamming-loss and Precision, the methods based on PS and the proposal by Spolaôr rank higher than the RPFS procedures. Two notable exceptions are: for Hamming-loss, the method Correlation Relevant ranks higher than PS IG and for the Precision, PS IG ranks eleventh.
2) The variation associated with the methods PS and Spolaôr are consistently larger than the variation associated with the RPFS procedures.

In Section 5.6 a summary of the results presented in Sections 5.4 and 5.5 is provided.

5.6 Conclusion

In the first section of Chapter 5, a motivation for the use of synthetic datasets in the evaluation of new procedures was provided. This was followed by a discussion of the methods available for generating synthetic multi-label datasets. The choice of the method proposed by Sandrock and Steel (2015) was motivated, and the properties of synthetic multi-label datasets were discussed. The scope of the empirical investigation was presented by summarising all the cases considered in this dissertation.

The experimental approach that was followed was detailed in the next section. Specific mention was made of the construction of the relevance matrix and the choice of thresholds used to determine relevance. The next section provided a discussion of the application of the proposed RPFS procedure based on MCA biplots. Section 5.3 was concluded by a section devoted to a discussion of the classification algorithms used in the empirical study.

The results were summarised and presented in Section 5.4. In the first two subsections, 5.4.1 and 5.4.2, the focus was on:

a) comparing the different fitted models, *i.e.* the model using the full set of features, the model that includes only the relevant features, the model that includes only the highest ranked feature from each feature group, and the model that includes the two highest ranking features from each feature group; and
b) comparing the different relevance measures, namely the correlation coefficient, IG, and ReliefF.

Section 5.4.1 presented the results for the SVM base classifier. The reduced models consistently performed better than the full model. When the reduced models are considered, the Relevant model performed better than the Highest 2 model, which in turn performed better than the Highest model. As for the relevance measures, the FS procedures based on the correlation coefficient and IG performed better than those based on ReliefF. It is also noted that the SVM classifier is sensitive to smaller values of density when the number of training instances is also fairly small.

The results for the XGBoost classifier were presented in Section 5.4.2 and these follow the same format as the results in Section 5.4.1. However, the results presented in this section were not as straightforward as for the SVM classifier. For example, for some of the evaluation measures, the FS procedures based on ReliefF ranked higher than those based on the correlation coefficient and IG for certain datasets. However, if the procedures are ranked over all 24 datasets using the Method of Pairwise Comparisons, one can conclude that the reduced models perform better than the full model. As is the case for the SVM classifier, for XGBoost the FS procedures based on the correlation coefficient and IG generally performed better than the procedures based on ReliefF. The model ReliefF Highest consistently ranked lowest of all the FS procedures. Unlike the SVM classifier, the XGBoost classifier is able to handle smaller values of densities irrespective of the number of training instances available.

The focus of Section 5.4.3 was to determine whether a particular classifier can be recommended. In general, the conclusion that could be drawn was that the procedures implementing the XGBoost classifier perform better when one considers either Hamming-loss or Precision, but that the procedures based on the SVM classifier perform better than those based on the XGBoost classifier for either One-error or Recall.

The performances of the FS procedures, regardless of classifier or relevance measure, improve for all four evaluation measures if the signal strength or the number of irrelevant features increases. The latter is counterintuitive and can possibly be explained by the fact that the number of training instances is increased as the proportion of irrelevant features increases. The influence of label dependence and choice of density vector on the performances of the procedures is more difficult to determine and the results presented in this regard are inconclusive.

In Section 5.5, the performances of the proposed RPFS procedures were compared to those of the established FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013).

When all 24 datasets are considered, the RPFS procedures ranked higher for One-error and Recall, but the procedures based on PS and Spolaôr typically ranked higher for Hamming-loss and Precision. Results for the RPFS procedures are generally less variable than those for the other FS procedures.

A similar pattern was observed in the rankings of the procedures for groupings of the data based on signal strength, number of irrelevant features, or number of training instances. When the datasets are grouped based on label dependence, the RPFS procedures rank highest for One-error and Precision. The results for Hamming-loss and Precision are not consistent. If the labels are not correlated, the methods based on RPFS, particularly the methods based on the correlation coefficient as relevance measure, rank fairly high, but if some degree of label correlation is present, the rankings of methods based on PS improve significantly. This same pattern is observed when the datasets are grouped based on different vectors of densities. The RPFS procedures rank higher for One-error and Recall, but for Hamming-loss, PS and Spolaôr perform better. For Precision, the rankings of methods based on PS improve significantly if the vector of densities entries are varied.

For the majority of the 24 synthetic datasets, the FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al.* (2013) include fewer features in the reduced models than the RPFS procedures presented in this dissertation.

In the next chapter, the results of this dissertation will be summarised, and some further research ideas and recommendations will be suggested.

CHAPTER 6

CONCLUSION

6.1 Summary

The first chapter of this dissertation served as a brief introduction to the field of multi-label data, including an overview of multi-label classification and multi-label FS. The problem addressed in the dissertation can be summarised as follows: FS procedures aim to identify features that are relevant and/or important, providing information when labels have to be assigned to new data instances. In the multi-label context, this is a more complicated problem than in the single-label case.

Before the problem could be addressed, the notion of feature relevance as proposed by Sandrock and Steel (2016) was introduced. Thereafter, a two-stage FS procedure was suggested. During the first step, features are grouped into unique, non-overlapping feature groups. The features need to be grouped based on the information that these features provide to the classifier. MCA biplots allow for such a grouping to be made. Features that plot close to each other on the MCA biplot will provide similar information. The row points obtained from the MCA biplot are plotted in a new RPFS plot to visualise the feature groups.

The second step of the proposed FS procedure involves the selection of features from the feature groups created during the first step. The relevance measures used to determine the feature groups provide an implicit ranking of the features within each of the groups. This ranking is utilised in the second step to select features.

Chapter 2 discussed general aspects of multi-label classification and the evaluation measures that are used to evaluate multi-label classification techniques. The discussion of the evaluation measures focussed on example-based measures as well as measures that make use of the label ranking to evaluate the performance of a procedure. Multi-label classification approaches were presented and described in Sections 2.4 and 2.5. Section 2.4 focussed on the popular problem transformation methods of BR, LP, and RPC, while Section 2.5 focussed on the popular algorithm adaptation method of ML-kNN. Finally, the two base classifiers used in this study, namely the SVM and XGBoost, were described.

In Chapter 3, several general approaches followed in the field of FS were presented. These approaches were discussed with specific attention to the methods proposed by Sandrock and Steel (2016) and Spolaôr *et al.* (2013). Finally, a new RPFS method was proposed that utilises the ability of MCA biplots to group features into homogeneous groups. The rest of the dissertation was devoted to the application of the proposed approach to a benchmark dataset (in Chapter 4) and to synthetic datasets (in Chapter 5).

The chapter focussing on the application of the RPFS procedure on the *Emotions* benchmark dataset started with an introduction devoted to the *Emotions* multi-label benchmark dataset. Implementation of the proposed technique on the *Emotions* dataset was presented in Section 4.4. This included a detailed discussion of the construction of the relevance matrix with specific reference to the three relevance measures used and to the way in which the thresholds for relevance are determined. The specific choices made regarding the BR problem transformation method and the SVM and XGBoost classifiers employed in the dissertation were also explained. The results of RPFS based on the SVM and XGBoost classifiers were presented, followed by a comparison of RPFS implementing these two classifiers in Section 4.5.

Four models were compared, namely

- the full model;
- the model including all the features that are deemed to be relevant;
- the model including only the highest ranked feature from each group; and
- the model including the two highest ranked features from each group.

For the last two models, the features are ranked according to the relevance measures. The performance of the model including all the relevant features is similar to that of the full model. The FS procedures that include fewer features do not perform as well, but the results are still fairly competitive even though substantially fewer features are used.

Three relevance measures are included in this dissertation, namely the correlation coefficient, IG, and ReliefF. For the *Emotions* dataset there does not seem to be a substantial difference between the results obtained from these three relevance measures. However, the model that only includes the highest ranked feature using ReliefF and the XGBoost classifier performed better than the same model based on the correlation coefficient and IG.

In Section 4.5.3 the performance of the FS procedures based on the two classifiers were compared. For the evaluation measures Hamming-loss and Precision the FS procedures based on the XGBoost classifier performed better than those based on the SVM classifier, but the FS procedure using the SVM performed better than those implementing XGBoost for Recall. The results for One-error were inconclusive, with the FS procedures for the SVM as classifier performing only slightly better than those that use XGBoost as classifier.

In Chapter 5 a motivation for the use of synthetic datasets in the evaluation of new FS procedures was provided. A critical review of the methods available for generating synthetic multi-label datasets was presented and the rationale for choosing the method proposed by Sandrock and Steel (2015) was given. The section on generating synthetic multi-label data was completed by a discussion of the properties of such datasets. The cases included in this dissertation, *i.e*. the structure of the 24 synthetic datasets that are used in the empirical study, were summarised in a single table.

The experimental approach followed in Chapter 5 is similar to that followed in Chapter 4. The results for the FS procedures implementing the SVM classifier were presented in Section 5.4.1. The reduced models consistently perform better than the full model. When the reduced models are considered, the Relevant model performed better than the Highest 2 model, which in turn performed better than the model Highest. As for the relevance measures, the FS procedures based on the correlation coefficient and IG performed better than those based on ReliefF.

In Section 5.4.2 the results based on the XGBoost classifier were presented. For certain evaluation measures, the FS procedures based on ReliefF ranked higher than those based on the correlation coefficient and IG for certain datasets. As is the case for the SVM classifier, for XGBoost the FS procedures based on the correlation coefficient and IG generally performed better than those procedures based on ReliefF. The model ReliefF Highest was consistently ranked lowest of all the FS procedures. If the procedures are ranked over all 24 datasets using the Method of Pairwise Comparisons, the reduced models generally performed better than the full model.

The influence of the classifier on the performances of the FS procedures was investigated in Section 5.4.3. For either Hamming-loss or Precision, the XGBoost classifier performs better, but for One-error and Recall, the FS procedures based on the SVM perform better.

Regardless of the classifier or the relevance measure used, the performances of the FS procedures improve for all four evaluation measures if the signal strength and number of irrelevant features increase. The latter does not make sense, and it is critical to consider the fact that the number of training instances is dependent on the proportion of irrelevant features in this empirical study. Therefore, the improved performance should be ascribed to the increase in the number of training instances available. The results are inconclusive for the investigation into the influence of label dependence and the choice of density vector on the performance of the FS procedures.

In the last section of Chapter 5, the performance of the proposed RPFS procedure was compared to that of the established FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013). When all datasets are considered, the RPFS procedures rank higher for One-error and Recall, but the procedures based on PS and Spolaôr typically rank higher for Hamming-loss and Precision. The performance of the RPFS procedures is generally associated with less variation than the performance of the other FS procedures.

This same pattern was observed in the rankings of the procedures for groupings of the data based on signal strength, the number of irrelevant features, or the number of training instances. If the datasets are grouped based on label dependence, the RPFS procedures rank highest for One-error and Precision. The results for Hamming-loss and Recall were not as consistent as they were for the other groupings. When no correlation is present between the labels, the methods based on RPFS, especially the methods based on the correlation coefficient, rank high, but if some degree of label correlation is present, the ranking of methods based on PS improve. When the datasets are grouped based on different vectors of densities, the same pattern is observed. The RPDS procedures outranked the other procedures for One-error and Recall, but for Hamming-loss, PS and Spolaôr performed better. For Precision, the rankings of methods based on PS improved significantly if the vector of densities entries are varied.

For most of the 24 synthetic datasets, the FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al.* (2013) include fewer features in their corresponding reduced models than the RPFS procedures.

Some ideas for future research directions will be presented in next section.

6.2 Research contributions

This dissertation focused on introducing a novel approach to FS in the multi-label context. The proposed method uses the methodology of MCA biplots to perform the initial grouping of the features based on the relevance matrix obtained using the three relevance measures, namely the correlation coefficient, IG, and ReliefF. The MCA biplot enables one to group features together that provide similar information (*i.e.* that lie close together on the biplot). A RPFS plot is constructed using the row points obtained during MCA to visualise the feature groups. During the second stage, one can then utilise the inherent ranking abilities of the relevance measures to identify the features that rank highest in each of the feature groups.

The first contribution of this research is based on the fact that the methods CA, MCA, and biplots are well-known and powerful statistical techniques. Biplots are used extensively to provide graphical representations of complex multivariate datasets, but to date biplots have not been used for FS.

Secondly, the use of MCA biplot methodology to perform FS provides the practitioner with a visual representation of the associations between the features and the labels. To date such a visualisation has not been available.

The proposed RPFS procedures allow for features that provide similar information to be grouped together and features can be ranked according to importance within these groupings. This ranking allows the user to determine the importance of different features. In medical fields, for example, where some medical procedures are very expensive, insight into the importance of these medical procedures could potentially lead to large savings in time and costs.

The fourth contribution is based on the fact that there has been an increase in interest in multi-label classification. This, combined with the diverse practical applications, implies that the development of a new FS technique could have significant practical implications.

Finally, the use of the procedure to generate artificial multi-label datasets proposed by Sandrock and Steel (2016) allows the researcher to control the number of relevant features, feature importance, and label densities. To date, these options have not been available to researchers who are interested in comparing different multi-label FS procedures.

6.3 Future research recommendations

While the FS procedures proposed in this dissertation performed well in the empirical investigations, there are some further areas that need to be addressed.

6.3.1 Extending the simulation study

A first recommendation would be to extend the simulation study to include for example a larger number of features in the synthetic datasets. Some real-world multi-label datasets contain a large number of features. For example, consider the *Toxic Comment Classification Challenge* (Kaggle, 2017). This challenge is aimed at building a model that is able to classify and detect different types of toxicity like threats, obscenity, insults, and identity-based hate from comments made online. These comments consist of text data from which features can be extracted using methods like Bag of Words or Topic Modelling. The resulting dataset could contain thousands of features. Some of the benchmark datasets from the text domain also have a large number of features, for example the dataset *tmc2007* which is based on aviation reports has 49 060 features.

Only four evaluation measures were included in the empirical evaluation in this dissertation. In future studies it would be interesting to include more evaluation measures. More varied ratios of relevant to irrelevant features should also be investigated in an expanded simulation study.

If a large number of relevant features are included in the synthetic datasets, it could lead to a very large number of feature groups being formed. If the majority of these feature groups only includes a single feature, a large number of features will be included by the FS procedure. For this reason, the criterion used to create feature groups may need to be adjusted.

6.3.2 Relaxing the criterion for creating feature groups

The method proposed in this dissertation assumes that features that are plotted *close* together in the RPFS plot are similar, and that these features will provide similar information to the classifier. A decision has to be made on: How close is close? A strict definition of close is that features are considered close only if they plot directly on top of each other. Such a strict constraint would imply a more conservative (larger) number of feature groups.

Relaxing the criterion to group features into groups if they lie within a certain distance from each other should be investigated. Revisiting the RPFS plot for the *Emotions* dataset based on ReliefF as relevance measure one could, for example, decide to group the feature groups circled in orange together in Figure 6.1.

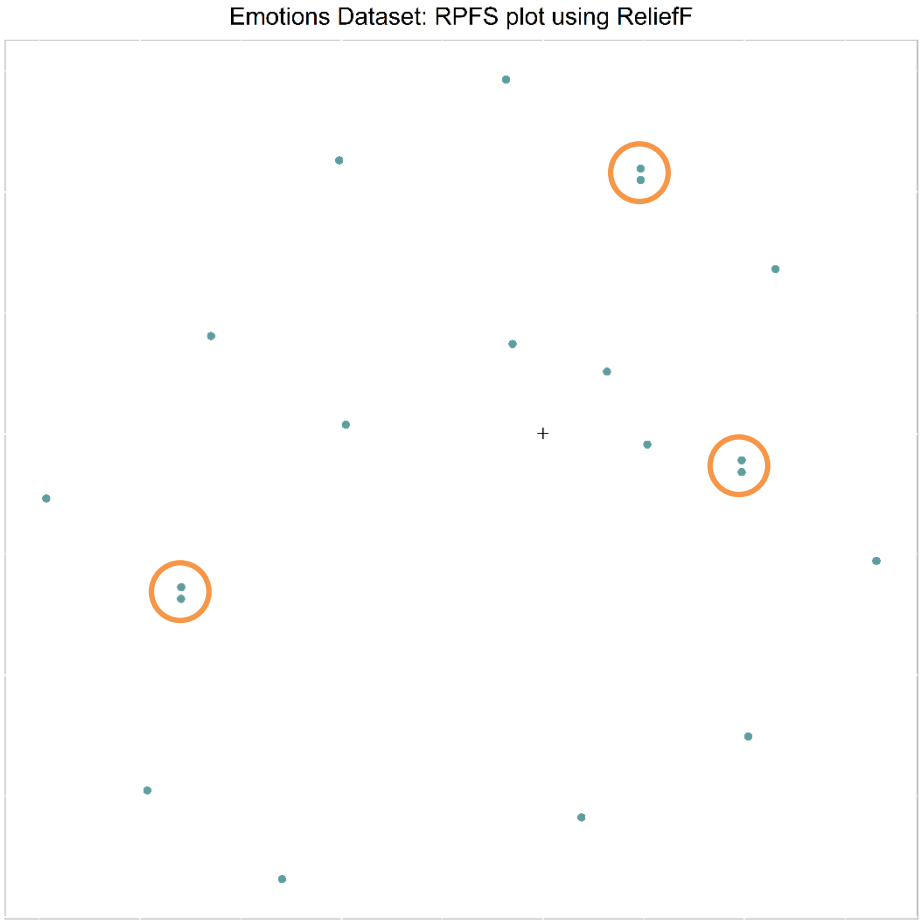


Figure 6.1 RPFS plot using ReliefF as relevance measure (revisited).

A formal distance criterion, which specifies a measure that determines when feature groups are deemed to be close together, can be investigated. This criterion could then be included as a user-specified parameter in the RPFS procedure. The Mahalanobis distance would be a good candidate for such a measure.

6.3.3 Exploiting the full potential of MCA biplots

While this research does utilise some aspects of MCA biplot methodology, there is still scope for further investigation into the full potential of exploiting MCA biplots for multi-label FS. This should ideally be done in cooperation with an expert in this broad field.

Since the FS procedure performs relatively well on benchmark and synthetic data, it would be interesting to evaluate its performance on a larger, real-world example such as the Alzheimer dataset.

6.3.4 Alzheimer data

Research by the Alzheimer's Disease Neuroimaging Initiative (ADNI) suggests that one in three Americans over the age of 65 is affected by Alzheimer's disease (AD). The disease is currently ranked as the sixth overall leading cause of death in the U.S.A., ranking even higher for older Americans (ADNI, n.d.).

This dataset is receiving increasing interest from researchers in the field of multi-label classification. Refer to Cheng *et al*. (2015), Cheng *et al*, (2018), and Zhang *et al*. (2018). The effective treatment, prevention, and cure of AD have become of critical importance, as the American population is aging due to the baby boomer generation that are all of retirement age.

In October 2004 the ADNI started researching the use of biomarkers for the diagnosis of AD. These biomarkers include blood tests, cerebrospinal fluid tests, Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET) scans. The research is aimed at finding more sensitive and accurate methods to detect AD at earlier ages and to mark the disease's progress through biomarkers, brain scans, and genetic profiles. The project was later expanded to include the use of biomarkers to detect AD at a pre-dementia stage. In the early diagnosis of Alzheimer's, the goal is to find discriminative brain regions from biomarkers which include brain images. These brain regions will be considered as the labels and the biomarkers as the features.

In this particular context, the importance of FS is clear: medical testing and imaging are extremely expensive procedures in the U.S.A., and techniques that are able to identify which of these techniques are most important, *i.e.* provide the most information in the diagnosis of AD could result in significant savings with respect to cost and time to diagnosis.

6.3.5 Grouping of the labels

In light of the phenomenon growth in unstructured text data (Moore, 2018), it is imperative that classification and FS techniques that are developed are able to deal with large text datasets. An example of an application where the technique proposed in this dissertation could potentially add value, is the categorisation of text messages into categories (labels) in Customer Service call centres.

The customers of Mobile Network Operators (MNO) in South Africa are notoriously disloyal. It is critical for these companies to monitor their customers' experiences in real time. One aspect that they specifically want to monitor, is a customer's experience with their Customer Services call centre. Customers with good experiences are less likely to switch to a new MNO when their contract ends.

Once a customer has contacted the MNO's call centre, the customer receives a text message from the MNO asking for them to provide feedback in the form of a text message elaborating on their experience. The text messages then need to be analysed in order for the MNO to provide better service to their customers.

An executive of the MNO is tasked with identifying categories (labels) such as, for example, poor call centre agent training, reception issues, limited product offering, *etc*. Each text response is automatically categorised into one or more of these categories using a classifier. The resulting dataset suffers from poor prediction due to its sparse and unbalanced structure. Preliminary investigation shows that combining categories or labels could lead to an improved model. Industry currently relies on domain experience to combine categories, since there is very little theoretical research to guide these decisions.

A technique that groups these categories into fewer categories in a justifiable manner could lead to better prediction and improve the MNO's ability to better understand their customers' experiences with their call centres.

In this dissertation the focus has been on forming feature groups to aid FS, but the proposed technique can easily be adapted to provide groupings of the labels instead. These label groups that are formed will be more defendable than groupings created subjectively based on domain experience.

While MCA biplots allow for the visualisation of high-dimensional data, there are other visualisation methods which could be applied in a similar fashion.

6.3.6 Opportunity for comparative studies

One such method is t-Distributed Stochastic Neighbour Embedding (t-SNE) proposed by Van der Maaten and Hinton (2008). The method models each high-dimensional object in a low-dimensional space (two or three dimensions) in a manner such that similar objects are modelled by points that are close to one another, and dissimilar objects by more distant points.

Van der Maaten and Hinton (2008) show that t-SNE is capable of retaining the local structure of the data while providing important information about grouping. The comparison of the FS procedure proposed in this dissertation with an established technique such as t-SNE would be an interesting future research direction.

REFERENCES

Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B. & Swami, A. 1992. An interval classifier for database mining applications. In *Proceedings of the 18th International Conference on Very Large Databases, Vancouver, Canada* (pp. 560 – 573).

Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., García, S., Sánchez, L. & Herrera, F. 2011. KEEL data mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-valued Logic and Soft Computing*, Volume 17 (pp. 255 – 287).

Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernandez, J.C. & Herrera, F. 2009. KEEL: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing*, Volume 13 (3) (pp. 307 – 318).

Alzheimer's Disease Neuroimaging Initiative. [n.d.] [Online] Available: http://adni.loni.usc.edu/ [2018, December 14].

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K. & Müller, K-R. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research*, Volume 11 (pp. 1803 – 1831).

Barros, R.C., Cerri, R., Freitas, A. & de Carvalho, A. 2013. Probabilistic clustering for hierarchical multi-label classification of protein functions. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Prague, Czech Republic* (pp. 385 – 400).

Barutcuoglu, Z., Schapire, R.E. & Troyanskaya, O.G. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics*, Volume 22 (7) (pp. 830 – 836).

Bernardini, F.C., Da Silva, R.B., Rodovalho, R.M. & Meza, E.B.M. 2014. Cardinality and density measures and their influence to multi-label learning methods. *Journal of the Brazilian Society on Computational Intelligence – Learning and Nonlinear Models*, Volume 12 (1) (pp. 53 - 71).

Bi, J., Bennett, K., Embrects, M., Breneman, C. & Song, M. 2003. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, Volume 3 (pp. 1229 – 1243).

Blockeel, H., Raedt, L.D. & Ramon, J. 1998. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning, San Francisco, CA, USA* (pp. 55 – 63).

Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. 2012. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, Volume 24 (pp. 483 – 519).

Boutell, M.R., Lou, J., Shen, X. & Brown, C.M. 2004. Learning multi-label scene classification. *Pattern Recognition*, Volume 37 (9) (pp. 1757 – 1771).

Bradu, D. & Gabriel, K.R. 1978. The biplot as a diagnostic tool for models of two-way tables. *Technometrics*, Volume 20 (1) (pp. 47 – 68).

Braytee, A., Liu, W., Catchpoole, D.R. & Kennedy, P.J. 2017. Multi-label feature selection using correlation information. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM - 2017), Singapore, Singapore* (pp. 1649 – 1656).

Brownlee, J. 2016. *A gentle introduction to XGBoost for applied machine learning*. [Online] Available: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ [2019, March 24].

Bühlmann, P. & Hothorn, T. 2007. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, Volume 22 (4) (pp. 477 – 505).

Cai, Y., Yang, M., Yang, G. & Yin, H. 2015. ReliefF-based multi-label feature selection. *International Journal of Database Theory and Application*, Volume 8 (4) (pp. 307 – 318).

Cao, L. 2016. Data science and analytics: a new era. *International Journal of Data Science and Analytics*, Volume 1 (1) (pp. 1 – 2).

Charte, F., Rivera, A.J., Charte, D., del Jesus, M.J. & Herrera, F. 2018. Tips, guidelines and tools for managing multi-label datasets: The `mldr.datasets` R package and the `Cometa` data repository. *Neurocomputing*, Volume 289 (pp. 68 – 85).

Chen, T. & Guestrin, C. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD Conference of Knowledge Discovery and Data Mining (KKD – 2016), San Francisco, CA, USA.*

Chen, W., Yan, J., Zhang, B., Chen, Z. & Yang, Q. 2007. Document transformation for multi-label feature selection in text categorization. In *Proceedings of the 7th IEEE International Conference on Data Mining Workshops (ICDMW – 2007), Omaha, NE, USA.*

Cheng, B., Liu, M. & Zhang, D. 2015. Multimodal multi-label transfer learning for early diagnosis of Alzheimer's disease. In: Zhou L., Wang L., Wang Q. & Shi Y. (eds), *Machine Learning in Medical Imaging (MLMI - 2015) Munich, Germany*, Lecture Notes in Computer Science, Volume 9352 (pp. 238 – 245).

Cheng, B., Liu, M., Zhang, D., Shen, D. & Alzheimer's Disease Neuroimaging Initiative. 2018. Robust multi-label transfer feature learning for early diagnosis of Alzheimer's Disease. *Brain Imaging and Behavior*, Volume 1 (pp. 1 – 16).

Cherman, E.A., Metz, J. & Monard, M.C. 2010. A simple approach to incorporate label dependency in multi-label classification. In *Proceedings of the Mexican International Conference on Artificial Intelligence: Advances in Soft Computing (MICAI – 10), Pachuca, Mexico* (pp. 33 – 43).

Cherman, E.A., Metz, J. & Monard, M.C. 2012. Incorporating label dependency into the binary relevance framework. *Expert Systems with Applications*, Volume 39 (2) (pp. 1647 – 1655).

Chou, S. & Hsu, C.L. 2005. MMDT: a multi-valued and multi-labelled decision tree classifier for data mining. *Expert Systems with Applications*, Volume 28 (pp. 799 – 812).

Clare, A. & King, R.D. 2001. Knowledge discovery in multi-label phenotype data. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), Freiburg, Germany* (pp. 42 – 53).

Cox, T.F. & Cox, M.A.A. 2000. *Multidimensional scaling*. London: Chapman and Hall/CRC.

Crammer, K. & Singer, Y. 2003. A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, Volume 3 (pp. 1025 – 1058).

Csima, E. 2014. Contemporary Mathematics 111 Class notes. Kentucky University [Online]. Available: https://www.ms.uky.edu/~csima/ma111/VotingLecture5.pdf [2018, October 10].

Data Science [n.d.]. [Online] Available: https://www.datarobot.com/wiki/data-science/ [2019, March 24].

De Comité, F., Gilleron, R. & Tommasi, M. 2003. Learning multi-label alternating decision trees from texts and data. In: Perner, P. & Rosenfeld, A. (eds), *Machine Learning and Data Mining in Pattern Recognition (MLDM - 2003), Leipzig, Germany*, Lecture Notes in Computer Science, Volume 2734 (pp. 35 – 49).

Dekel, O. & Shamir, O. 2010. Multiclass-multilabel classification with more classes than examples. In *JMLR Workshop and Conference Proceedings, Sardinia, Italy*, Volume 9 (pp. 137 – 144).

Dembczyński, K., Waegeman, W., Cheng, W. & Hüllermeier, E. 2010. On label dependence in multi-label classification. In *Second international workshop on learning from multi-label data (MLD - 2010), Haifa, Israel*, in conjunction with ICML/COLT 2010.

Dembczyński, K., Waegeman, W., Cheng, W. & Hüllermeier, E. 2012. On label dependence and loss minimization in multi-label classification. *Machine Learning*, Volume 88 (1-2) (pp. 5 – 45).

Demšar, J. 2010. Algorithms for subsetting values with Relief. *Machine Learning*, Volume 78 (3) (pp.421 – 428).

Dendamrongvit, S., Vateekul, P. & Kubat, M. 2011. Irrelevant features and imbalanced classes in multi-label text-categorization domains. *Intelligent Data Analysis*, Volume 15 (pp. 843 – 859).

Dimitrovski, I., Kocev, D., Loskovska, S. & Džeroski, S. 2012. Hierarchical classification of diatom images using ensembles of predictive clustering trees. *Ecological Informatics*, Volume 7 (1) (pp. 19 – 29).

Doquire, G. & Verleysen, M. 2013. Mutual information-based feature selection for multi-label classification. *Neurocomputing*, Volume 122 (pp. 148 – 155).

Dreyfus, G. & Guyon, I. 2006. Assessment methods. In Guyon, I., Gunn, S., Nikravesh, M. & Zadeh, L. (eds.) *Feature extraction – Foundations and applications*. New York: Springer (pp. 65 – 88).

Elisseeff, A. & Weston, J. 2001. A kernel method for multi-labelled classification. In *Proceedings of the 14th International Conference on Neural Information Processing Systems (NIPS): Natural and Synthetic Pages, Vancouver, Canada,* (pp. 681 – 687).

Farnsworth, P.R. 1958. *The social psychology of music*. New York: The Dryden Press.

Freund, Y. 1995. Boosting a weak learning algorithm by majority. *Information and Computation*, Volume 121 (2) (pp. 256 – 285).

Freund, Y. & Schapire, R.E. 1997. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, Volume 55 (pp. 119 – 139).

Friedman, J., Hastie, T. & Tibshirani, R. 2000. Additive logistic regression: a statistical view of boosting (with discussion). *Annals of Statistics*, Volume 28 (pp. 337 – 407).

Fürnkranz, J., Hüllermeier, E., Mencia, E.L. & Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Machine Learning*, Volume 73 (2) (pp.133 – 153).

Gabriel, K.R. 1971. The biplot graphical display of matrices with application to principal component analysis. *Biometrika*, Volume 58 (pp. 453 – 467).

Gabriel, K.R. 1981. Biplot. In S. Kotz, N.L. Johnson and C. Reads (eds.), *Encyclopaedia of Statistical Sciences,* Volume 1 (pp. 262 – 265).

Gabriel, K.R. & Zamir, S. 1979. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, Volume 21 (pp. 489 – 498).

Gandhi, R. 2018. Boosting algorithms: AdaBoost, Gradient Boosting and XGBoost [Online]. Available: https://hackernoon.com/boosting-algorithms-adaboost-gradient-boosting-and-xgboost-f74991cad38c [2019, February 10].

Ghamrawi, N. & McCallum, A. 2005. Collective multi-label classification. In *Proceedings of the 2005 ACM Conference on Information and Knowledge Management (CIKM - 2005), Bremen, Germany* (pp. 195 – 200).

Godbole, S. & Sarawagi, S. 2004. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD - 2004), Sydney, Australia,* (pp. 22 – 30).

Gower, J.C. 1990. Three dimensional biplots. *Biometrika,* Volume 77 (pp. 773 – 785).

Gower, J.C. 1992. Generalized biplots. *Biometrika,* Volume 79 (pp. 475 – 493).

Gower, J.C. & Hand, D.J. 1996. *Biplots*. London: Chapman & Hall.

Gower, J.C. & Harding, S. 1988. Nonlinear biplots. *Biometrika,* Volume 75 (pp. 445 – 455).

Gower, J.C., Lubbe, S. & Le Roux, N.J. 2011. *Understanding biplots*. Chichester: John Wiley & Sons, Inc.

Greenacre, M.J. 1984. *Theory and applications of correspondence analysis*. London: Academic Press.

Greenacre, M.J. 2007. *Correspondence analysis in practice – Second Edition*. Boca Raton: Chapham & Hall/CRC.

Greenacre, M.J. 2010. *Biplots in practice*. Madrid: BBVA Foundation.

Gu, Q., Li, Z. & Han, J. 2011. Correlated multi-label feature selection. In *Proceedings of the 2011 ACM International Conference on Information and Knowledge Management (CIKM – 2011), Glasgow, Scotland, UK* (pp. 1087 – 1096).

Gupta, P. & Anand, A. 2013. Multi label classification using label clustering. In *Proceedings of the 1st Indian Workshop on Machine Learning (ITT - 2013), Kanpur, India.*

Guyon, I. & Elisseeff, A. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research,* Volume 3 (pp. 1157 – 1182).

Han, H., Huang, M., Zhang, Y., Yang, X. & Feng, W. 2019. Multi-label learning with label specific features using correlation information. *IEEE Access,* Volume 7 (pp. 11474 – 11484).

Hastie, T., Tibshirani, R. & Friedman, J. 2009. *The elements of statistical learning – Data Mining, inference, and prediction – Second Edition*. New York: Springer.

Kaggle. 2017. [Online] *Toxic comment classification challenge*. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview [2019, March 26].

Kashef, S., Nezamabadi-pour, H. & Nikpour, B. 2018. Multilabel feature selection: A comprehensive review and guiding experiments. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Volume 8 (2) (pp. 1 – 29).

Kaushik, S. 2016. Introduction to feature selection methods with an example (or how to select the right variables?). *Analytics Vidhya – Learning everything about analytics*. Available: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/ [2019, February 8].

Kira, K. & Rendell, L. 1992. A practical approach to feature selection. *In Proceedings of the 9th International Workshop on Machine Learning, Aberdeen, Scotland, UK* (pp. 249 – 256).

Kohavi, R. & John, G.H. 1997. Wrappers for feature selection. *Artificial Intelligence*, Volume 97 (1 – 2) (pp. 273 – 324).

Konenenko, I. 1994. Estimating attributes: Analysis and extensions of Relief. In *Proceedings of the 7th European Conference on Machine Learning (ECML – 1994), Catania, Italy* (pp. 171 – 182).

Kong, D., Ding, C.H.Q., Huang, H. & Zhao, H. 2012. Multi-label ReliefF and F-statistic feature selections for image annotation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA* (pp. 2352 – 2359).

Krier, C., François, D., Wertz, V. & Verleysen, M. 2006. Feature scoring by mutual information for classification of mass spectra. In *Proceedings of the 7th International FLINS Conference, Genova, Italy* (pp. 557 – 564).

Lastra G., Luaces O., Quevedo J.R. & Bahamonde A. 2011. Graphical feature selection for multilabel classification tasks. In: Gama J., Bradley E. & Hollmén J. (eds), *Advances in Intelligent Data Analysis X (IDA - 2011),* Lecture Notes in Computer Science, Volume 7014 (pp. 246 – 257).

Lee, J. & Kim, D-W. 2013. Feature selection for multi-label classification using multivariate mutual information. *Pattern Recognition Letters,* Volume 34 (3) (pp. 349 – 357).

Le Roux, N. & Lubbe, S. 2013. `UBbipl`: Understanding biplots: Data sets and functions. R package version 3.0.4. http://wiley.com/go/biplots.

Li, T. & Ogihara, M. 2003. Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval, Washington D.C., USA* (pp.564 – 574).

Lipton, Z.C. 2017. The mythos of model interpretability. In *Proceedings of the 2016 ICML Workshop on Human Interpretability in Machine Learning, New York. USA.*

Logan, B. & Salomon, A. 2001. A music similarity function based on signal analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME – 2001), Tokyo, Japan.*

Luaces, O., Díez, J., Del Coz, J.J., Barranquero, J. & Bahamonde, A. 2012. Synthetic datasets for sound experimental evaluation of multilabel classifiers. M-L Group, Artificial Intelligence Center, Universidad de Oviedo and Gijón.

Luo, X. & Zincar-Heywood, A.N. 2005. Evaluation of two systems on multi-class multi-label document classification. In *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems, Saratoga Springs, NY, USA* (pp. 161 – 169).

Madjarov, G., Kocev, D., Gjorgjevikj, D. & Džeroski, S. 2012. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition,* Volume 45 (9) (pp. 3084 – 3104).

Marr, B. 2016. *What is the difference between artificial intelligence and machine learning?* [Online]. Available: https://www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/#6c34cf2a2742 [2019, March 24].

Melo, A. & Paulheim, H. 2019. Local and global feature selection for multi-label classification with binary relevance. *Artificial Intelligence Review*, Volume 51 (1) (pp. 33 – 60).

Moor, J. 2006. The Dartmouth College Artificial Intelligence Conference: The next fifty years. *AI Magazine*, Volume 27 (4) (pp. 87 – 89).

Moore, M.B. 2018. *The untapped potential of unstructured text*. [Online] Available: https://insidebigdata.com/2018/10/22/untapped-potential-unstructured-text/ [2019, March 26].

Noh, H.G., Song, M.S., & Park, S.H. 2004. An unbiased method for constructing multilabel classification trees. *Computational Statistics*, Volume 47 (pp. 149 – 164).

Olsson, J. & Oard, W. 2006. Combining feature selectors for text classification. In *Proceedings of the 2006 ACM International Conference on Information and Knowledge Management (CIKM – 2006), Arlington, VA, USA* (pp. 798 – 799).

Oman, S. 2009. Easily simulated multi-variate binary distributions with given positive and negative correlations. *Computational Statistics and Data Analysis*, Volume 53 (pp. 999 – 1005).

Pereira, R.B., Plastino, A., Zadrozny, B. & Merschmann, L.H. 2015. Information gain feature selection for multi-label classification. *Journal of Information and Data Management*, Volume 6 (1) (pp. 48 – 58).

Pereira, R.B., Plastino, A., Zadrozny, B. & Merschmann, L.H. 2018. Categorizing feature selection methods for multi-label classification. *Artificial Intelligence Review*, Volume 49 (pp. 57 – 78).

Pupo, O.G.R., Morell, C. & Soto, S.V. 2013. ReliefF-ML: an extension of ReliefF algorithm to multi-label learning. In: Ruiz-Shulcloper, J. & Sanniti di Baja, G. (eds.) *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, Volume 8259 (pp. 528 – 535).

Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T. & Zhang, H. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia, Augsburg, Germany* (pp. 17 – 26).

Qi, G., Hua, X., Rui, Y., Tang, J. & Zhang, H. 2009. Two-dimensional multi-label active learning with an efficient online adaptation model for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 31 (10) (pp. 1880 – 1897).

Quevedo, J.R., Bahamonde, A. & Luaces, O. 2007. A simple and efficient method for variable ranking according to their usefulness for learning. *Computational Statistics & Data Analysis*, Volume 52 (1) (pp. 578 – 595).

Rakotomamonjy, A. 2003. Variable selection using SVM-based criteria, *Journal of Machine Learning Research*, Volume 3 (pp. 1357 – 1370).

Read, J. 2008. A pruned problem transformation method for multi-label classification. In *Proceedings of the 2008 New Zealand Computer Science Research Student Conference (NZCSRS – 2008), Christchurch, New Zealand* (pp. 143 – 150).

Read, J. 2013. Multi-label classification [Online]. Available: https://users.ics.aalto.fi/jesse/talks/Multilabel-Part01.pdf [2019, February 16].

Read, J., Bifet, A., Holmes, G. & Pfahringer, B. 2012. Scalable and efficient multi-label classification for evolving data streams. *Machine Learning*, Volume 88 (pp. 243 – 272).

Read, J., Pfahringer, B. & Holmes, G. 2008. Multi-label classification using ensembles of pruned sets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM – 2008), Pisa, Italy* (pp. 995 – 1000).

Read, J., Pfahringer, B., Holmes, G. & Frank, E. 2009. Classifier chains for multi-label classification. In *Proceedings of the 20th European Conference on Machine Learning: Machine Learning and Knowledge Discovery in Databases, Bled, Slovenia* (pp. 254 – 269).

Read, J., Reutemann, P., Pfahringer, B. & Holmes, G. 2016. MEKA: A multi-label/multi-target extension to WEKA. *Journal of Machine Learning Research*, Volume 17 (pp. 1 – 5).

Reyes, O., Morell, C. & Ventura, S. 2013. ReliefF-ML: An extension of ReliefF algorithm to multi-label learning. In *Proceedings of the 18th Iberoamerican Congress (CIARP – 2013), Havana, Cuba* (pp. 528 – 535).

Reyes, O., Morell, C. & Ventura, S. 2015. Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. *Neurocomputing*, Volume 161 (pp.168 – 182).

Rivolli, A., Parker, L.C. & de Carvalho, A.C.P.L.F. 2017. Food truck recommendation using multi-label classification. In *Proceedings of the Portuguese Conference on Artificial Intelligence, Porto, Portugal*, (pp. 585 – 596).

Rivolli, A., Soares, C. & de Carvalho, A.C.P.L.F. 2018. Enhancing multilabel classification for food truck recommendation. *Expert Systems: Fourth Special Issue on Knowledge Discovery and Business Intelligence*, Volume 35 (4).

Robnik-Sikonja, M. & Konenenko, I. 1997. An adaptation of Relief for attribute estimation in regression. In *Proceedings of the 14th International Conference on Machine Learning (ICML – 1997), Nashville, TN, USA* (pp. 296 – 304).

Robnik-Sikonja, M. & Kononenko, I. 2003. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, Volume 53 (pp. 23 – 69).

Robnik-Sikonja, M. & Savicky, P. 2018. `CORElearn`: Classification, regression and feature evaluation. R package version 1.53.1. https://CRAN.R-project.org/package=CORElearn.

Rogati, M. & Yang, Y. 2002. High-performing feature selection for text classification. In *Proceedings of the 2002 ACM International Conference of Information and Knowledge Management (CIKM – 2002), McLean, VA, USA* (pp. 659 – 661).

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Cornell Aeronautical Laboratory, Psychological Review*, Volume 65 (pp.386 – 408).

Sandrock, G.K. 2013. *Multi-label feature selection with application to musical instrument recognition*. Published doctoral dissertation. University of Stellenbosch, Stellenbosch, South Africa.

Sandrock, G.K. & Steel, S.J. 2015. An algorithm for generating multi-label classification data. Technical report, University of Stellenbosch, Stellenbosch, South Africa.

Sandrock, G.K. & Steel, S.J. 2016. Probe variables for multi-label variable selection. Technical report, University of Stellenbosch, Stellenbosch, South Africa.

Schapire, R.E. 1990. The strength of weak learnability. *Machine Learning*, Volume 5 (2) (pp. 197 – 227).

Schapire, R.E. & Singer, Y. 2000. Boostexter: a boosting-based system for text categorization. *Machine Learning*, Volume 39 (pp. 135 – 168).

Schölkopf, B. & Smola, A.J. 2002. *Learning with kernels – Support vector machines, regularization, optimization, and beyond.* Cambridge: MIT Press.

Slavkov, I., Karcheska, J., Kocev, D., Kalajdziski, S. & Džeroski, S. 2013. Extending ReliefF for hierarchical multi-label classification. In *Workshop on New Frontiers in Mining Complex Patterns – European Conference on Machine Learning/Principles and Practice of Knowledge Discovery in Databases, Prague, Czech Republic* (pp. 156 – 167).

Slezak, D. & Ziarko, W. 2003. Attribute reduction in the Bayesian version of variable precision rough set model. *Electronic Notes in Theoretical Computer Science*, Volume 82 (pp. 263 – 273).

Solomonoff, R.J. 1985. The time scale of artificial intelligence. *Reflections on Social Effects – Human Systems Management*, Volume 5 (pp. 149 – 153).

Spolaôr, N., Cherman, E.A. & Monard, M.C. 2011. Using ReliefF for multi-label feature selection. In *Conferencia Latinoamericana de Informática, Quito, Ecuador* (pp. 960 – 975).

Spolaôr, N., Cherman, E.A., Monard, M.C. & Lee, H.D. 2012a. Filter approach feature selection methods to support multi-label learning based on ReliefF and Information Gain. In *Proceedings of the 21st Brazilian Conference on Advances in Artificial Intelligence, Curitiba, Brazil* (pp. 72 – 81).

Spolaôr, N., Cherman, E.A., Monard, M.C. & Lee, H.D. 2013. A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science*, Volume 292 (pp. 135 – 151).

Spolaôr, N., Lee, H.D., Takaki, W.S.R. & Wu, F.C. 2015. Feature selection for multi-label learning: A systematic literature review and some experimental evaluations. *International Journal of Computational Intelligence Systems*, Volume 8 (2) (pp. 3 – 15).

Spolaôr, N. & Monard, M.C. 2014. Evaluating Relieff-based multi-label feature selection algorithms. In *Proceedings of the 14th Edition of the Ibero-American Conference on Artificial Intelligence, Santiago de Chile, Chile,* Lecture Notes in Computer Science, Volume 8864 (pp 194 – 205).

Spolaôr, N., Monard, M.C. & Lee, H.D. 2012b. A systematic review to identify feature selection publications in multi-labeled data. ICMC Technical Report No. 374, University of São Paolo, São Paolo, Brazil, (pp. 31 – 51).

Spolaôr, N., Monard, M.C., Tsoumakas, G. & Lee, H.D. 2016. A systematic review of multi-label feature selection and a new method based on label correlation. *Neurocomputing*, Volume 108 (pp. 3 – 15).

Spyromitros, E., Tsoumakas, G. & Vlahavas, I. 2008. An empirical study of lazy multilabel classification algorithms. In *Proceedings of the 5th Hellenic Conference on Artificial Intelligence: Theories, Models and Applications, Syros, Greece* (pp. 401 – 406).

Stypka, J. 2015. Magpie Text Classification, Github repository. Available: https://github.com/inspirehep/magpie.

Thabtah, F.A., Cowling, P. & Peng, Y. 2004. MMAC: A new multi-class multi-label associative classification approach. In *Proceedings of the 4th IEEE International Conference on Data Mining, (ICDM 2004), Brighton, UK* (pp. 217 – 224).

Tomás, J.T., Spolaôr, N., Cherman, E.A. & Monard, M.C. 2014. A framework to generate synthetic multi-label datasets. *Electronic Notes in Theoretical Computer Science*, Volume 302 (pp. 155 – 176).

Trohidis, K., Tsoumakas, G., Kalliris, G. & Vlahavas, I. 2008. Multi-label classification of music into emotions. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA* (pp. 325-330).

Trohidis, K., Tsoumakas, G., Kalliris, G. & Vlahavas, I. 2011. Multi-label classification of music into emotions. *EURASIP Journal on Audio, Speech and Music Processing*, Volume 4.

Tsoumakas, G. & Katakis, I. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, Volume 3 (3) (pp. 1 – 13).

Tsoumakas, G., Katakis, I. & Vlahavas, I. 2010. Mining multi-label data. In O. Maimon and L. Rokach (eds). *Data Mining and Knowledge Discovery Handbook*. Berlin: Springer.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J. & Vlahavas, I. 2011. MULAN: A Java library for multi-label learning. *Journal of Machine Learning Research*, Volume 12 (pp. 2411 – 2414).

Tsoumakas, G. & Vlahavas, I. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML - 2007), Warsaw, Poland* (pp.406 – 417).

Tuv, E., Borisov, A., Runger, G. & Torkkola, K. 2009. Feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research*, Volume 10 (pp. 1341 – 1366).

Tuv, E., Borisov, A. & Torkkola, K. 2008. Ensemble-based variable selection using independent probes. *Computational Methods of Feature Selection*. Liu, H. And Motoda, H. (eds). Boca Raton, FL: Chapman & Hall/CRC.

Van der Maaten, L. & Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, Volume 9 (pp. 2579 – 2605).

Wei, Q., Yang, Z., Junping, Z. & Wang, Y. 2009. Semi-supervised multi-label learning algorithm using dependency among labels. In *Proceedings of the International Conference on Machine Learning and Computing (ICMLC – 2009), Darwin, Australia* (pp. 112 – 116).

Weng, L. 2017. How to explain the prediction of a machine learning model? [Online]. Available: https://lilianweng.github.io/lil-log/2017/08/01/how-to-explain-the-prediction-of-a-machine-learning-model.html [2019, July 12].

Weng, W., Lin, Y., Wu, S., Li, Y. & Kang, Y. 2018. Multi-label learning based on label-specific features and local pairwise label correlation. *Neurocomputing*, Volume 273 (pp. 385 – 394).

What is artificial intelligence? [n.d]. [Online]. Available: https://aws.amazon.com/machine-learning/what-is-ai/ [2019, March 24].

What is the genetic algorithm? [n.d.] [Online] Available: https://in.mathworks.com/help/gads/what-is-the-genetic-algorithm.html [2019, July 11].

Wieczorkowska, A., Synak, P. & Ras, Z. 2006. Multi-label classification of emotions in music. In *Intelligent Information Processing and Web Mining*. Berlin/Heidelberg: Springer. (pp. 307 – 315).

Yang, C. 2001. Music database retrieval based on spectral similarity. In *Proceedings of the International Symposium on Music Information Retrieval, Bloomington, IN, USA* (pp.37 – 38).

Yang, Y. & Pedersen, J.O. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML – 1997), Nashville, TN, USA* (pp. 412 – 420).

Younes, Z., Abdallah, F., Denoeux, T. & Snoussi, H. 2011. A dependent multilabel classification method derived from the k-nearest neighbor rule. *EURASIP Journal of Advances in Signal Processing*, Volume 2011 (pp. 1 – 14).

Yu, K., Yu, S. & Tresp, V. 2005. Multi-label informed latent semantic indexing. In *Proceedings of the 28th ACM SIGIR conference on Research and Development of Information Retrieval, Salvador, Brazil* (pp. 258 – 265).

Zhang, C., Adeli, E., Zhou, T., Chen, X. & Shen, D. 2018. Multi-layer multi-view classification for Alzheimer's disease diagnosis. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA* (pp. 4406 – 4413).

Zhang, L. & Duan, Q. 2019. A feature selection method for multi-label text based on feature importance. *Applied Sciences*, Volume 9 (4) (pp. 665 – 683).

Zhang, M.L., Peña, J.M. & Robles, V. 2009. Feature selection for multi-label naïve Bayes classification. *Information Sciences*, Volume 179 (pp. 3218 – 3229).

Zhang, M.L. & Zhou, Z.H. 2006. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, Volume 18 (10) (pp. 1338 – 1351).

Zhang, M.L. & Zhou, Z.H. 2007. ML-*k* NN: a lazy learning approach to multi-label learning. *Pattern Recognition*, Volume 40 (7) (pp. 2038 – 2048).

Zhang, Y. & Zhou, Z.H. 2010. Multilabel dimensionality reduction *via* dependence maximization. *ACM Transactions on Knowledge Discovery from Data*, Volume 4 (3) (pp. 1411 – 1421).

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., & Liu, H. 2011. Advancing Feature Selection Research, Technical Report-10-007, Arizona State University, Tempe, AZ, USA.

Zheng, Z., Wu, X. & Srihari, R. 2004. Feature selection for text categorization on imbalanced data. *ACM SIGKDD Explorations Newsletters*, Volume 6 (1) (pp. 80 – 89).

Zhu, S., Ji, X., Xu, W. & Gong, Y. 2005. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil* (pp. 274 – 281).

APPENDIX A

Feature Number	Feature Name
1	Mean_Acc1298_Mean_Mem40_Centroid
2	Mean_Acc1298_Mean_Mem40_Rolloff
3	Mean_Acc1298_Mean_Mem40_Flux
4	Mean_Acc1298_Mean_Mem40_MFCC_0
5	Mean_Acc1298_Mean_Mem40_MFCC_1
6	Mean_Acc1298_Mean_Mem40_MFCC_2
7	Mean_Acc1298_Mean_Mem40_MFCC_3
8	Mean_Acc1298_Mean_Mem40_MFCC_4
9	Mean_Acc1298_Mean_Mem40_MFCC_5
10	Mean_Acc1298_Mean_Mem40_MFCC_6
11	Mean_Acc1298_Mean_Mem40_MFCC_7
12	Mean_Acc1298_Mean_Mem40_MFCC_8
13	Mean_Acc1298_Mean_Mem40_MFCC_9
14	Mean_Acc1298_Mean_Mem40_MFCC_10
15	Mean_Acc1298_Mean_Mem40_MFCC_11
16	Mean_Acc1298_Mean_Mem40_MFCC_12
17	Mean_Acc1298_Std_Mem40_Centroid
18	Mean_Acc1298_Std_Mem40_Rolloff
19	Mean_Acc1298_Std_Mem40_Flux
20	Mean_Acc1298_Std_Mem40_MFCC_0
21	Mean_Acc1298_Std_Mem40_MFCC_1
22	Mean_Acc1298_Std_Mem40_MFCC_2
23	Mean_Acc1298_Std_Mem40_MFCC_3
24	Mean_Acc1298_Std_Mem40_MFCC_4
25	Mean_Acc1298_Std_Mem40_MFCC_5
26	Mean_Acc1298_Std_Mem40_MFCC_6
27	Mean_Acc1298_Std_Mem40_MFCC_7
28	Mean_Acc1298_Std_Mem40_MFCC_8
29	Mean_Acc1298_Std_Mem40_MFCC_9
30	Mean_Acc1298_Std_Mem40_MFCC_10
31	Mean_Acc1298_Std_Mem40_MFCC_11
32	Mean_Acc1298_Std_Mem40_MFCC_12
33	Std_Acc1298_Mean_Mem40_Centroid
34	Std_Acc1298_Mean_Mem40_Rolloff
35	Std_Acc1298_Mean_Mem40_Flux
36	Std_Acc1298_Mean_Mem40_MFCC_0

Feature Number	Feature Name
37	Std_Acc1298_Mean_Mem40_MFCC_1
38	Std_Acc1298_Mean_Mem40_MFCC_2
39	Std_Acc1298_Mean_Mem40_MFCC_3
40	Std_Acc1298_Mean_Mem40_MFCC_4
41	Std_Acc1298_Mean_Mem40_MFCC_5
42	Std_Acc1298_Mean_Mem40_MFCC_6
43	Std_Acc1298_Mean_Mem40_MFCC_7
44	Std_Acc1298_Mean_Mem40_MFCC_8
45	Std_Acc1298_Mean_Mem40_MFCC_9
46	Std_Acc1298_Mean_Mem40_MFCC_10
47	Std_Acc1298_Mean_Mem40_MFCC_11
48	Std_Acc1298_Mean_Mem40_MFCC_12
49	Std_Acc1298_Std_Mem40_Centroid
50	Std_Acc1298_Std_Mem40_Rolloff
51	Std_Acc1298_Std_Mem40_Flux
52	Std_Acc1298_Std_Mem40_MFCC_0
53	Std_Acc1298_Std_Mem40_MFCC_1
54	Std_Acc1298_Std_Mem40_MFCC_2
55	Std_Acc1298_Std_Mem40_MFCC_3
56	Std_Acc1298_Std_Mem40_MFCC_4
57	Std_Acc1298_Std_Mem40_MFCC_5
58	Std_Acc1298_Std_Mem40_MFCC_6
59	Std_Acc1298_Std_Mem40_MFCC_7
60	Std_Acc1298_Std_Mem40_MFCC_8
61	Std_Acc1298_Std_Mem40_MFCC_9
62	Std_Acc1298_Std_Mem40_MFCC_10
63	Std_Acc1298_Std_Mem40_MFCC_11
64	Std_Acc1298_Std_Mem40_MFCC_12
65	BH_LowPeakAmp
66	BH_LowPeakBPM
67	BH_HighPeakAmp
68	BH_HighPeakBPM
69	BH_HighLowRatio
70	BHSUM1
71	BHSUM2
72	BHSUM3

APPENDIX B

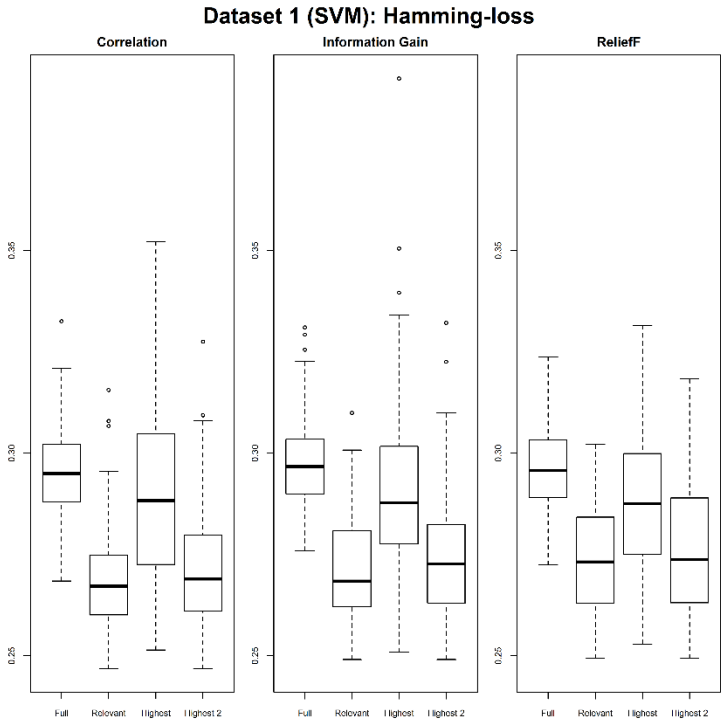


Figure B.1 Comparison of Hamming-loss using the SVM classifier: Dataset 1.

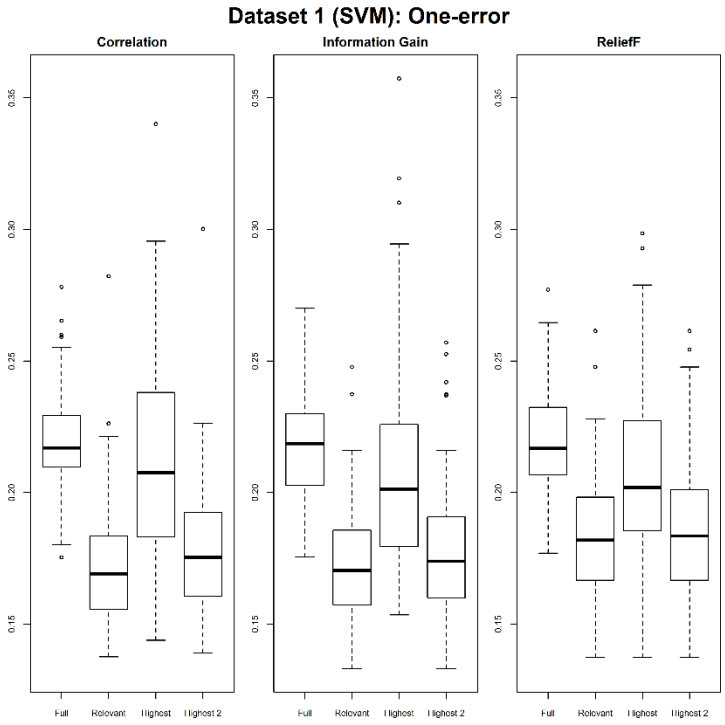


Figure B.2 Comparison of One-error using the SVM classifier: Dataset 1.

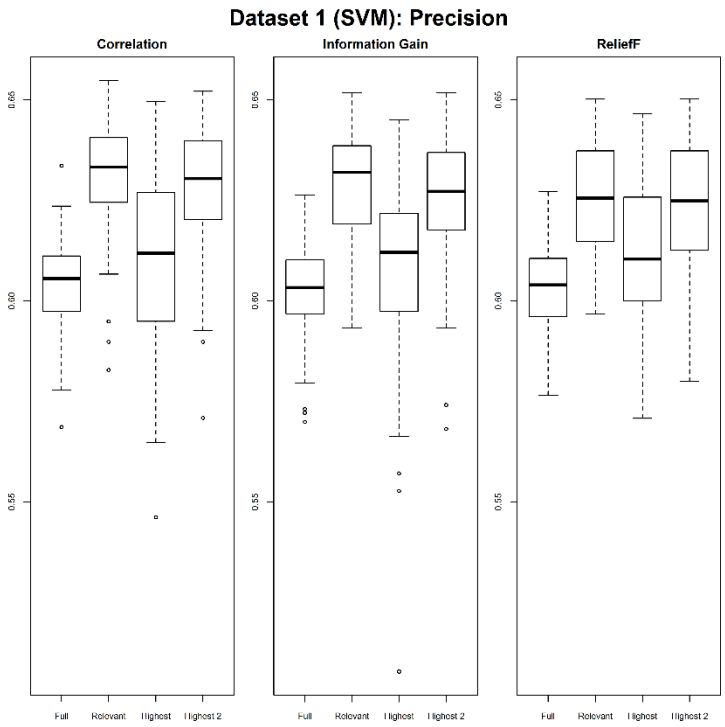


Figure B.3 Comparison of Precision using the SVM classifier: Dataset 1.

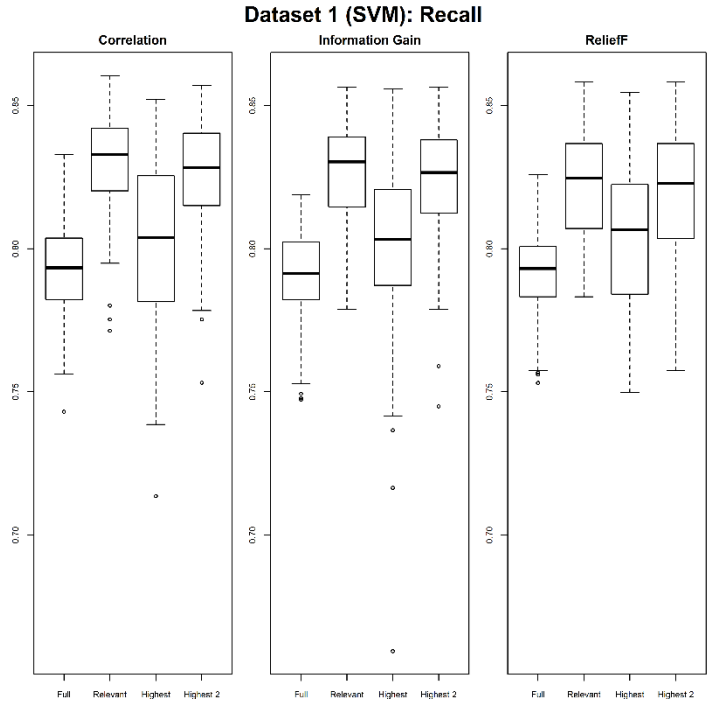


Figure B.4 Comparison of Recall using the SVM classifier: Dataset 1.

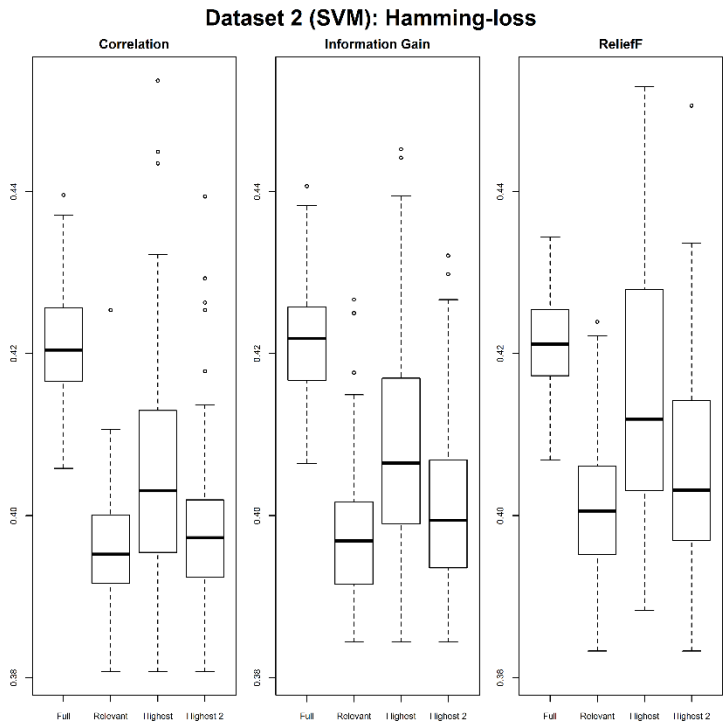


Figure B.5 Comparison of Hamming-loss using the SVM classifier: Dataset 2.

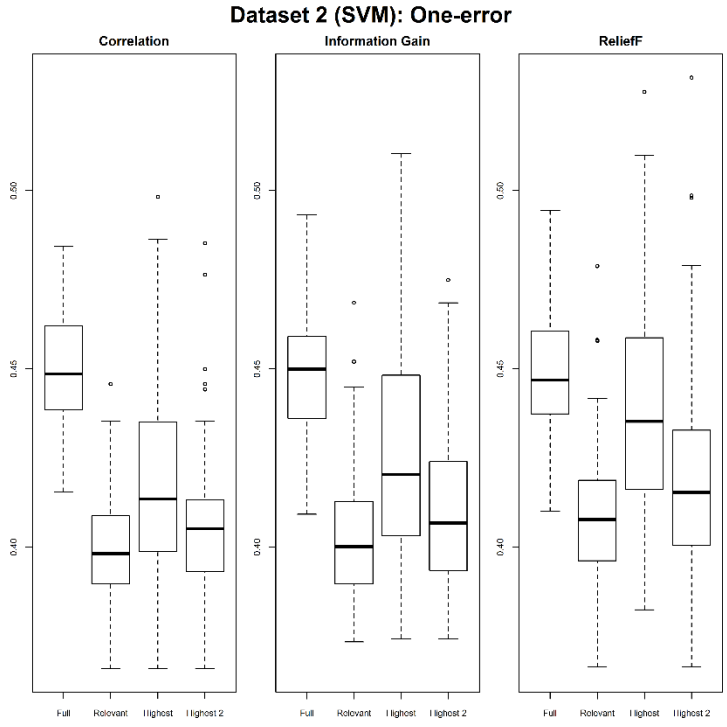


Figure B.6 Comparison of One-error using the SVM classifier: Dataset 2.

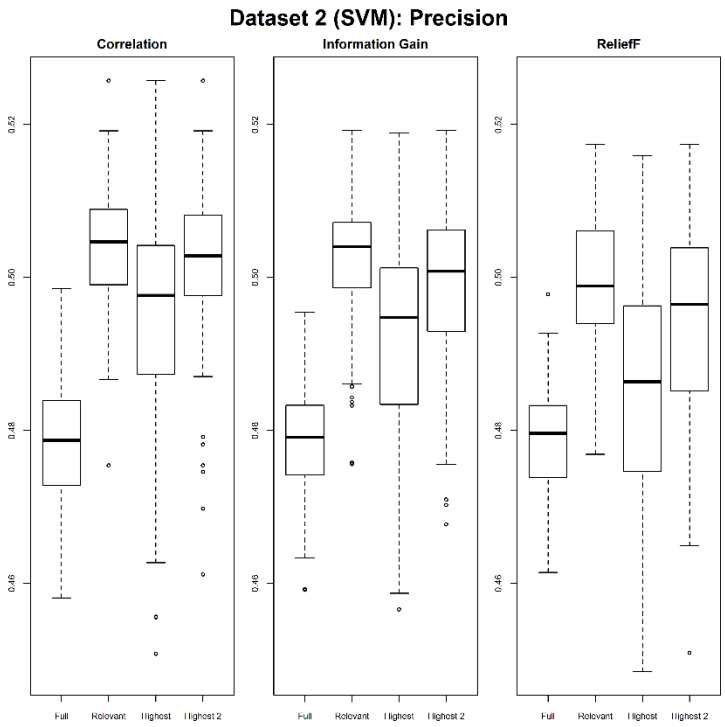


Figure B.7 Comparison of Precision using the SVM classifier: Dataset 2.

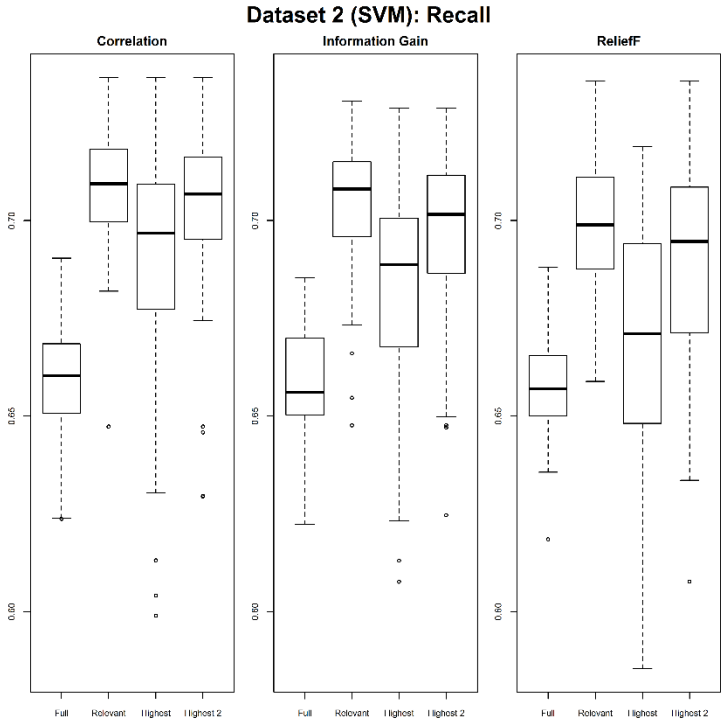


Figure B.8 Comparison of Recall using the SVM classifier: Dataset 2.

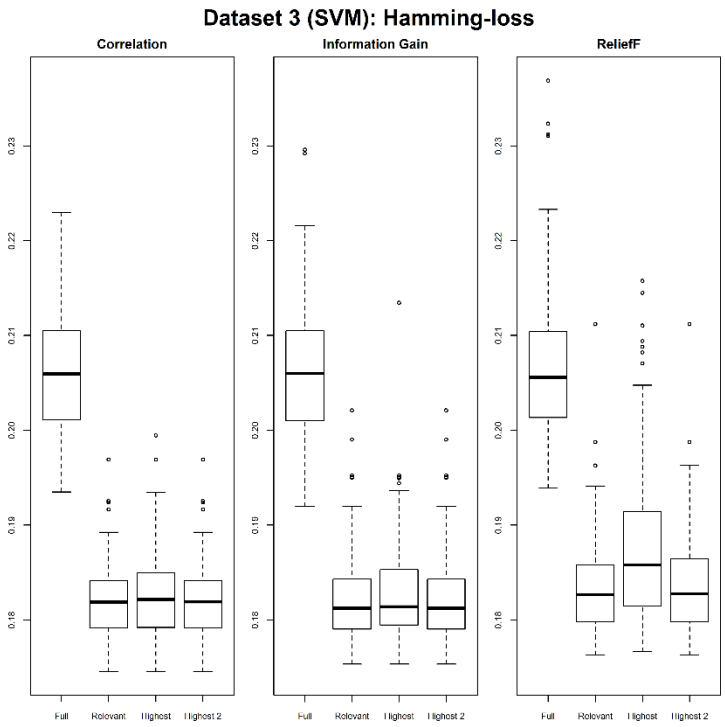


Figure B.9 Comparison of Hamming-loss using the SVM classifier: Dataset 3.

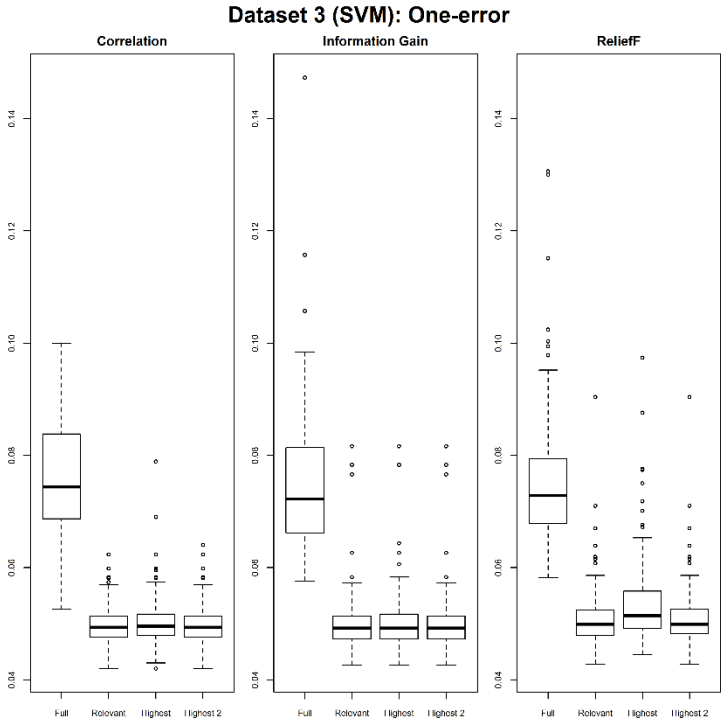


Figure B.10 Comparison of One-error using the SVM classifier: Dataset 3.

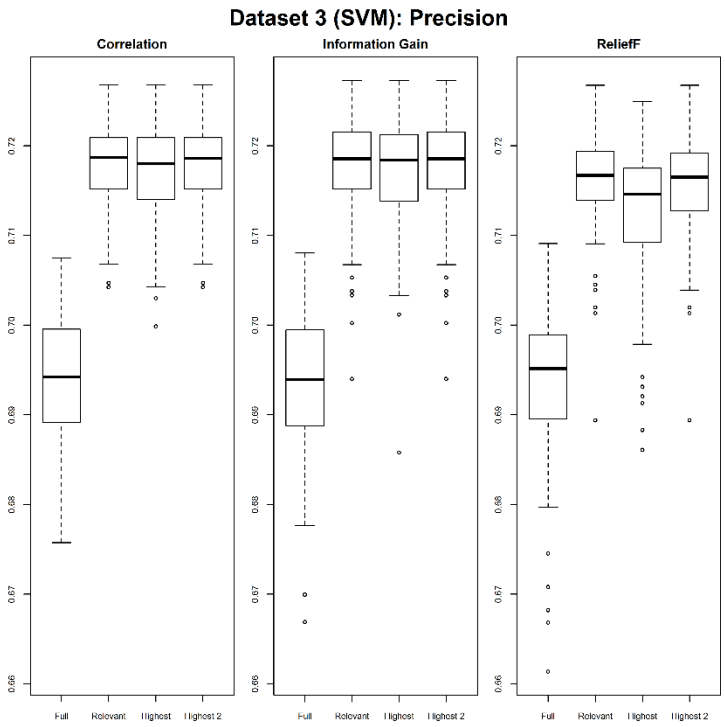


Figure B.11 Comparison of Precision using the SVM classifier: Dataset 3.

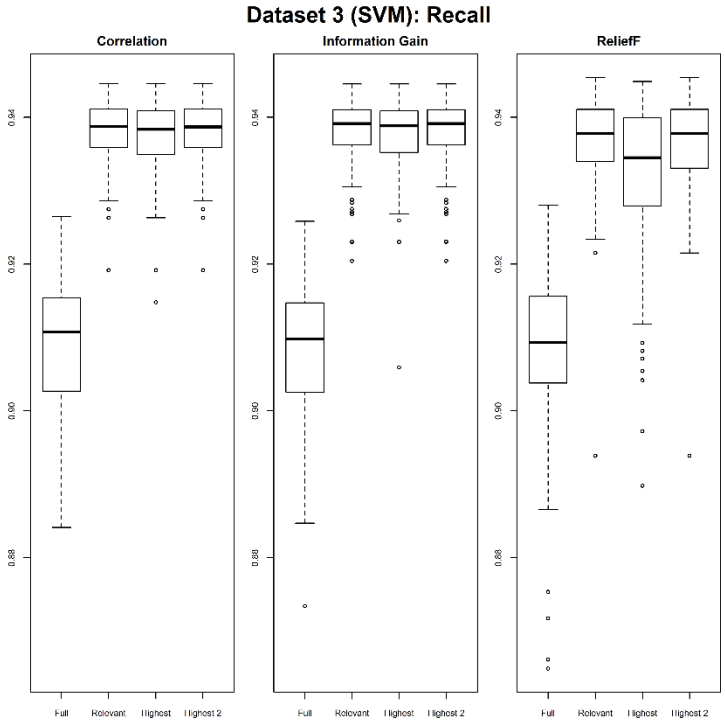


Figure B.12 Comparison of Recall using the SVM classifier: Dataset 3.

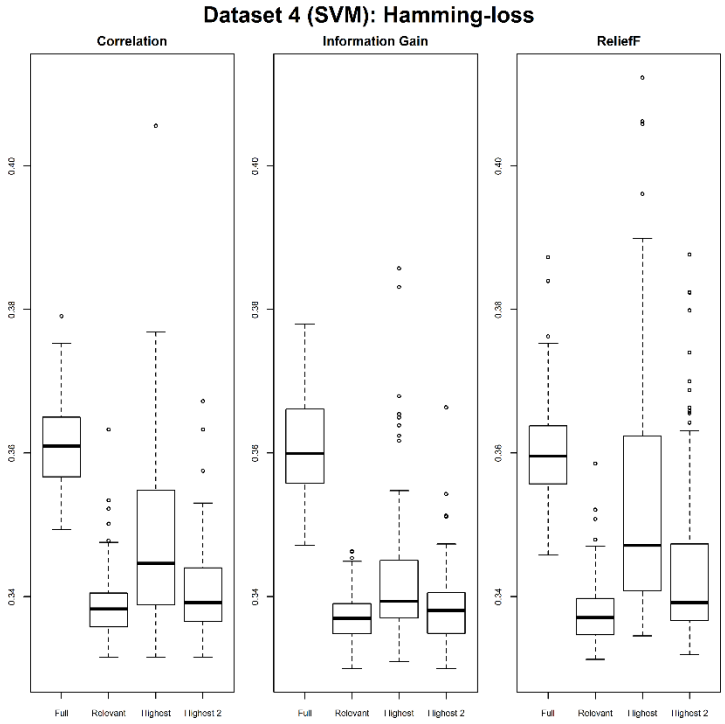


Figure B.13 Comparison of Hamming-loss using the SVM classifier: Dataset 4.

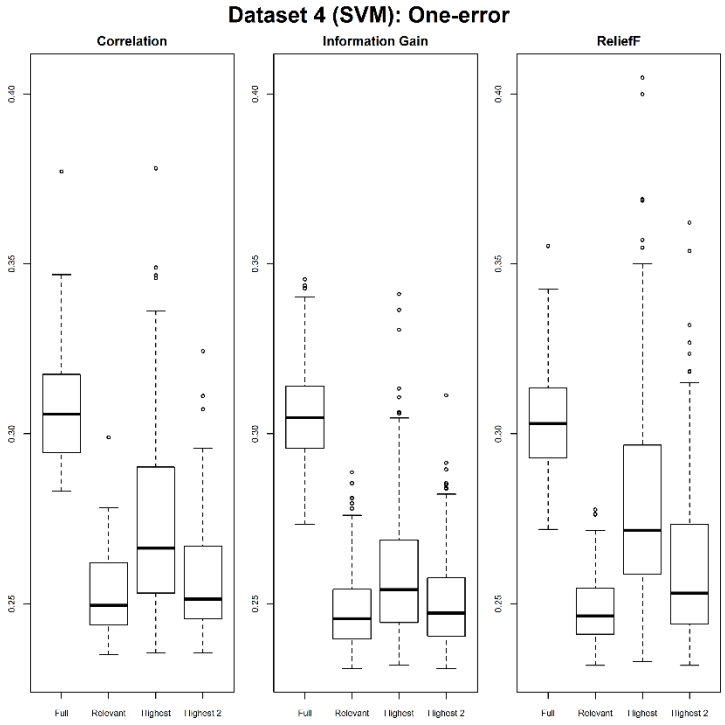


Figure B.14 Comparison of One-error using the SVM classifier: Dataset 4.

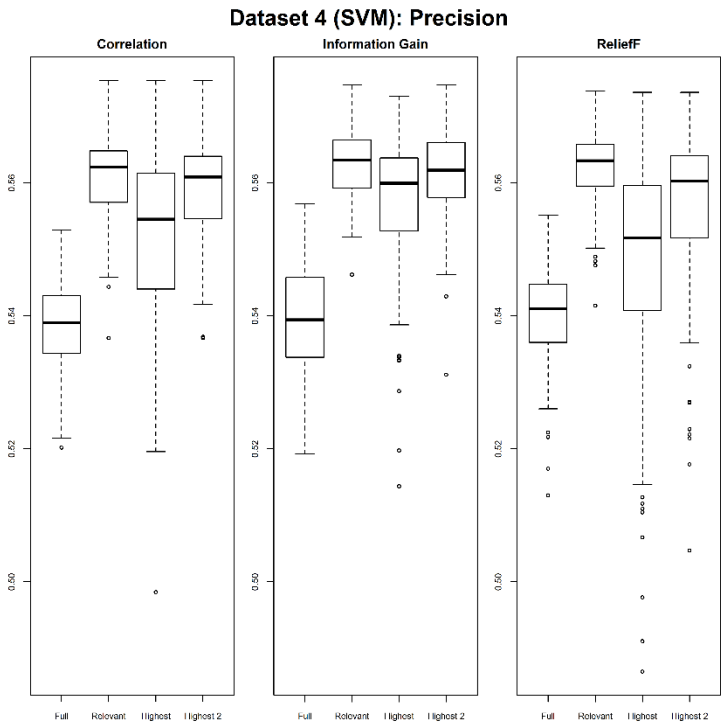


Figure B.15 Comparison of Precision using the SVM classifier: Dataset 4.

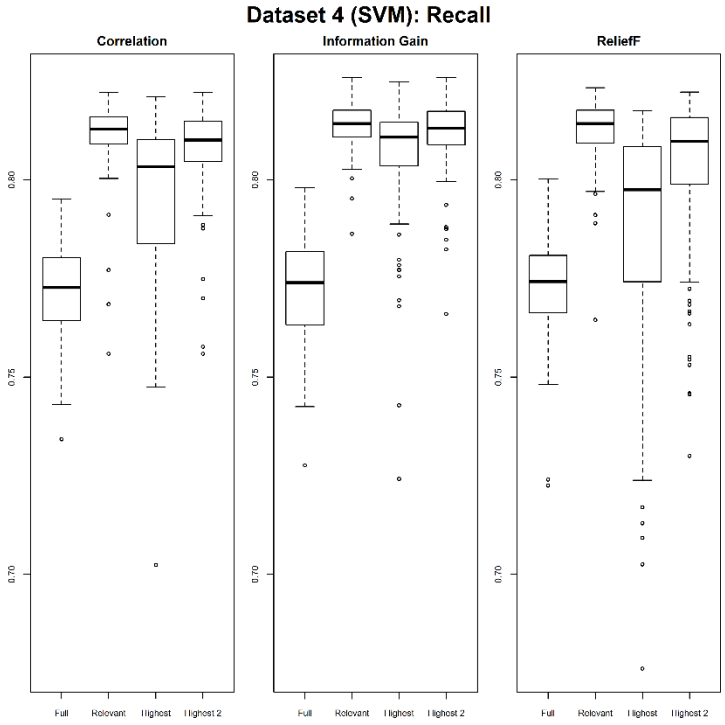


Figure B.16 Comparison of Recall using the SVM classifier: Dataset 4.

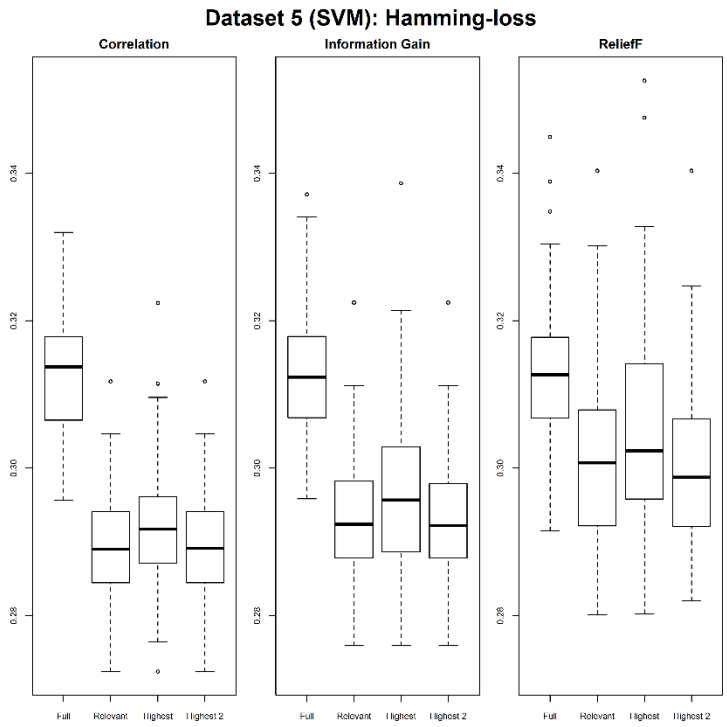


Figure B.17 Comparison of Hamming-loss using the SVM classifier: Dataset 5.

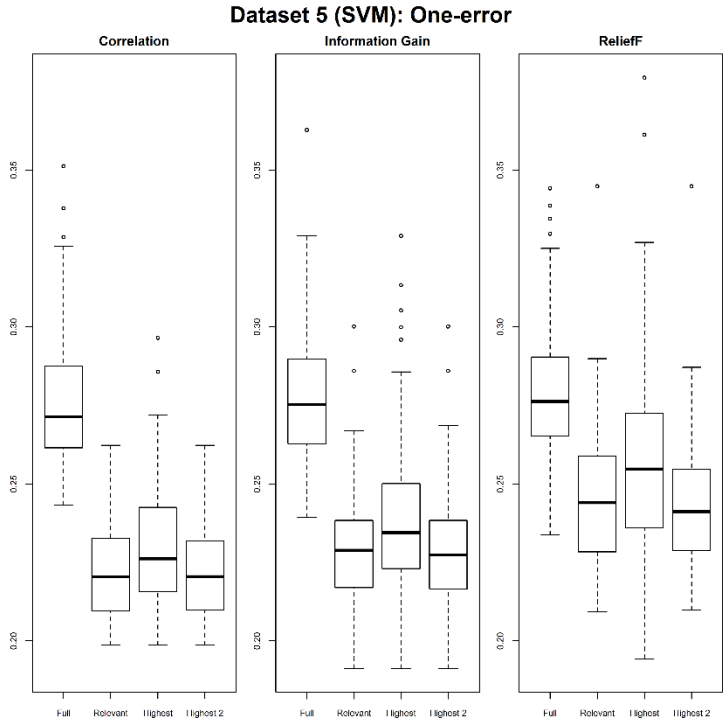


Figure B.18 Comparison of One-error using the SVM classifier: Dataset 5.

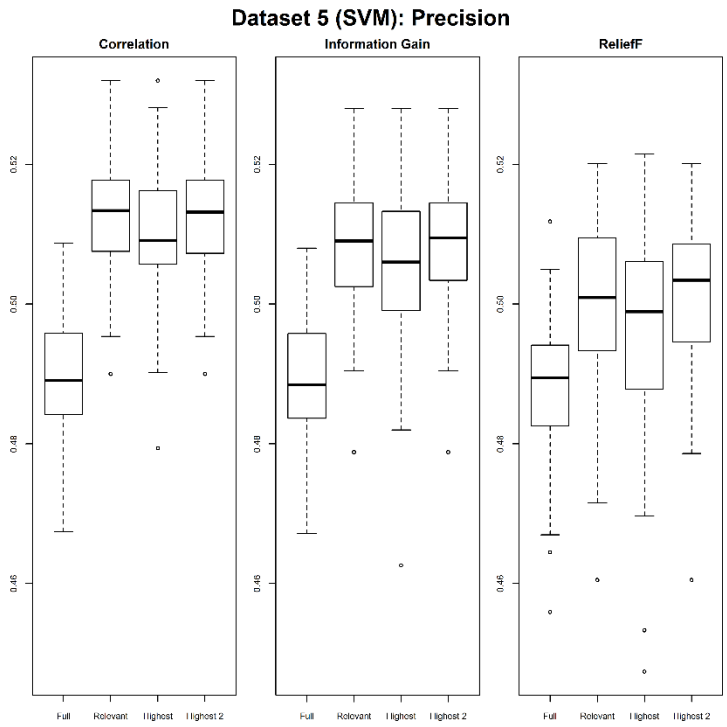


Figure B.19 Comparison of Precision using the SVM classifier: Dataset 5.

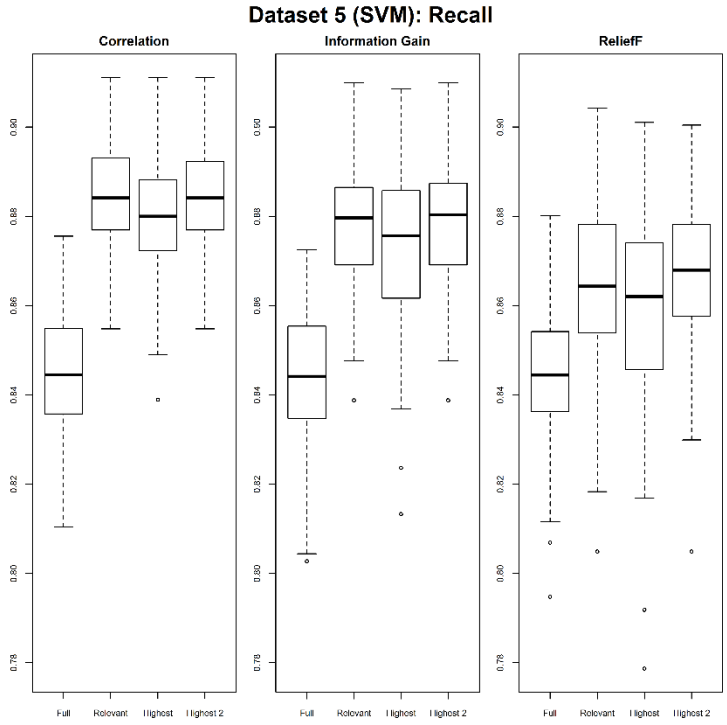


Figure B.20 Comparison of Recall using the SVM classifier: Dataset 5.

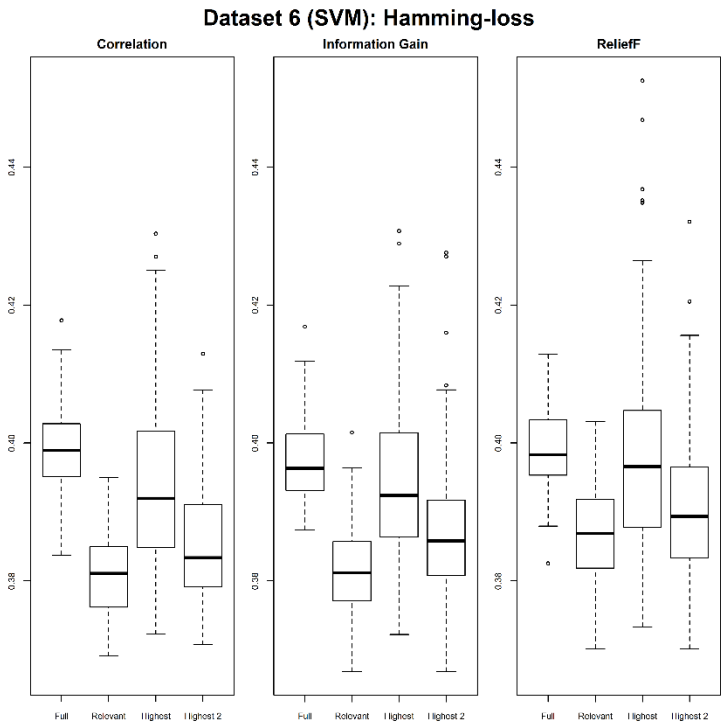


Figure B.21 Comparison of Hamming-loss using the SVM classifier: Dataset 6.

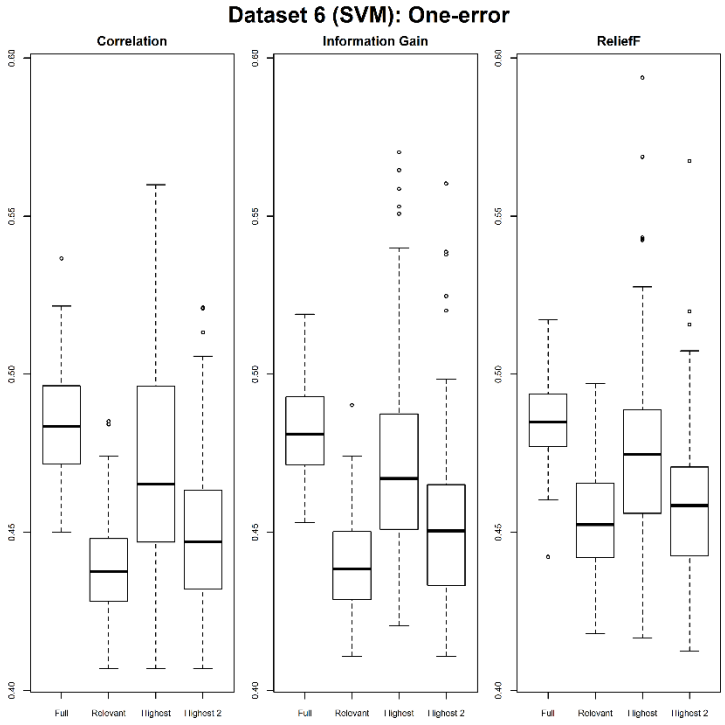


Figure B.22 Comparison of One-error using the SVM classifier: Dataset 6.

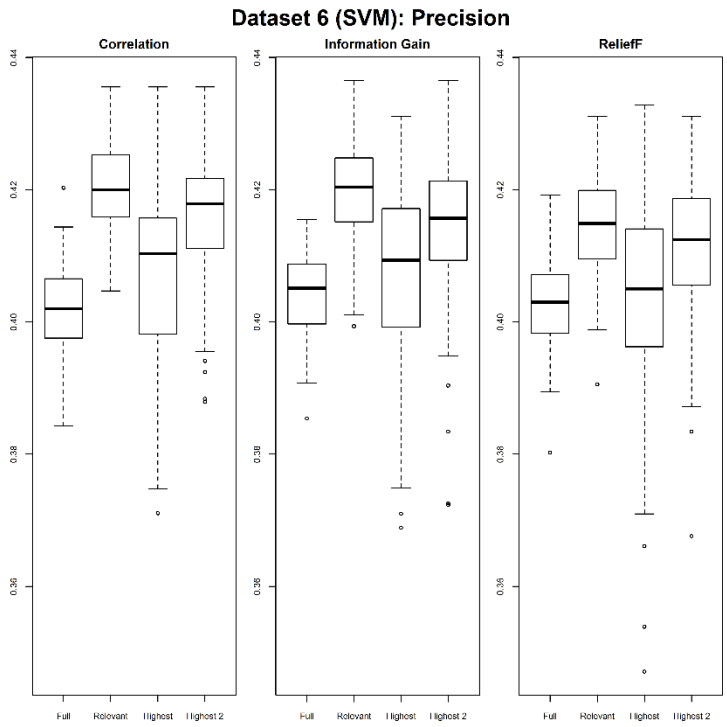


Figure B.23 Comparison of Precision using the SVM classifier: Dataset 6.

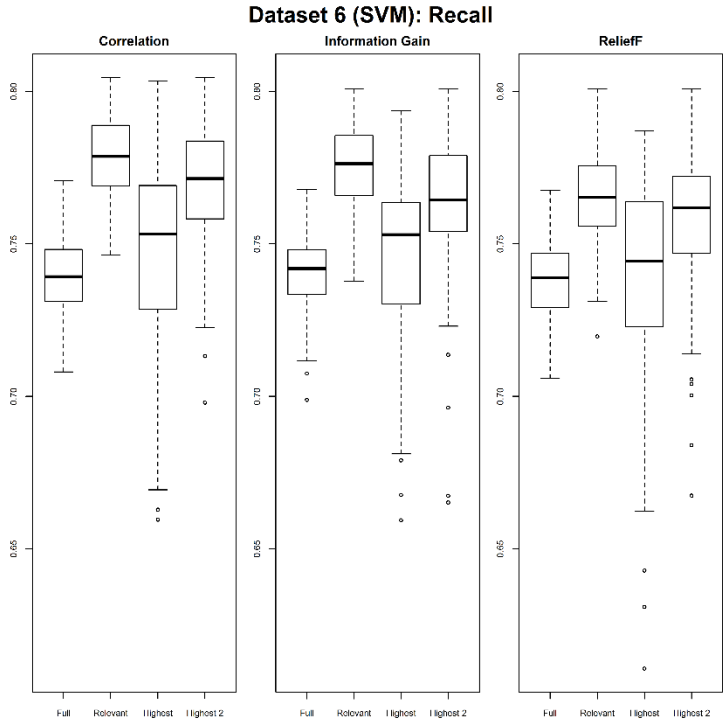


Figure B.24 Comparison of Recall using the SVM classifier: Dataset 6.

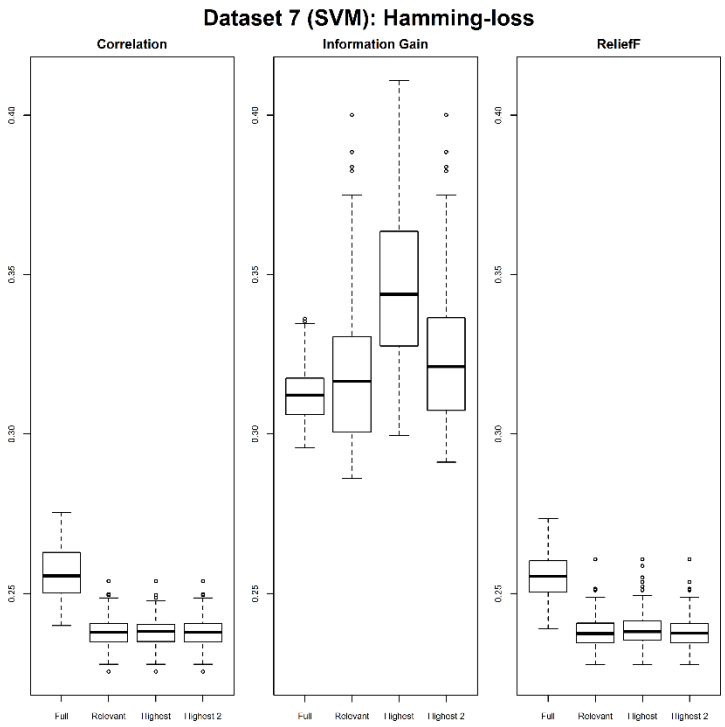


Figure B.25 Comparison of Hamming-loss using the SVM classifier: Dataset 7.

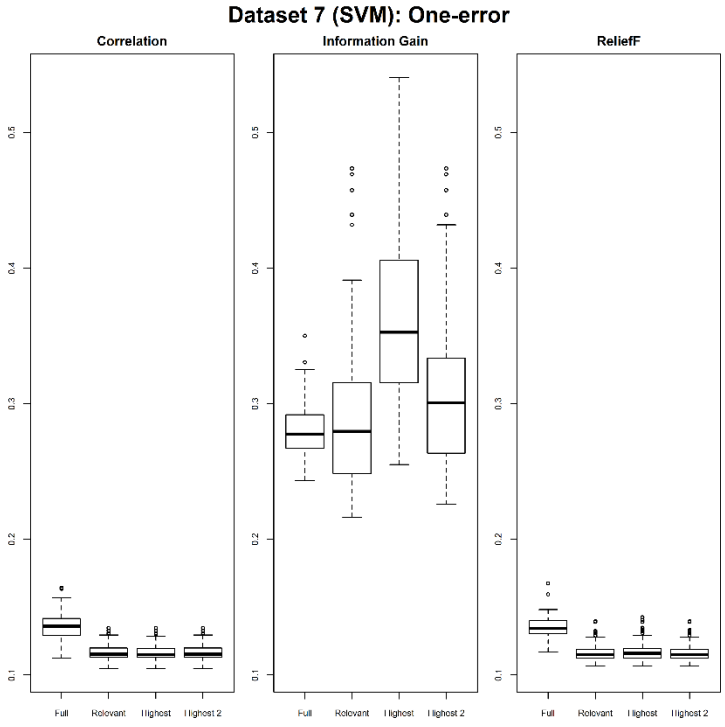


Figure B.26 Comparison of One-error using the SVM classifier: Dataset 7.

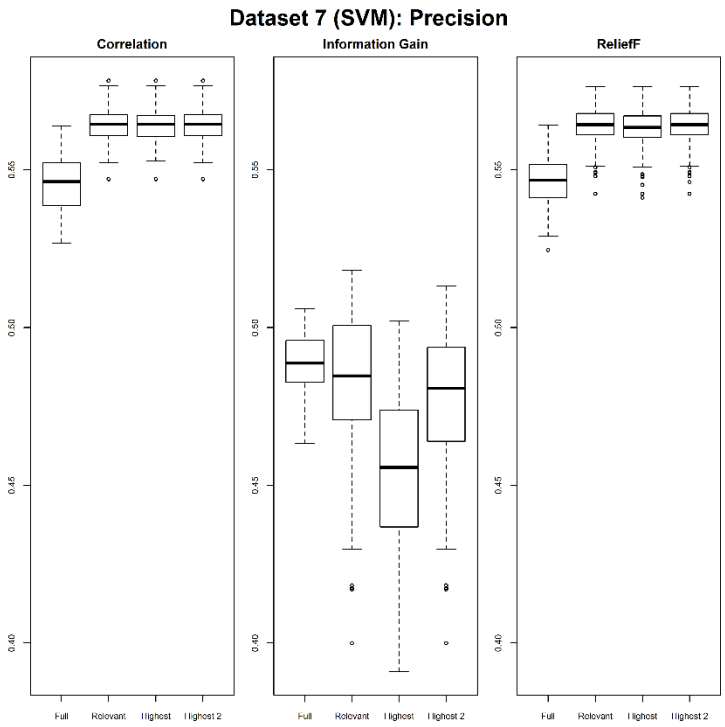


Figure B.27 Comparison of Precision using the SVM classifier: Dataset 7.

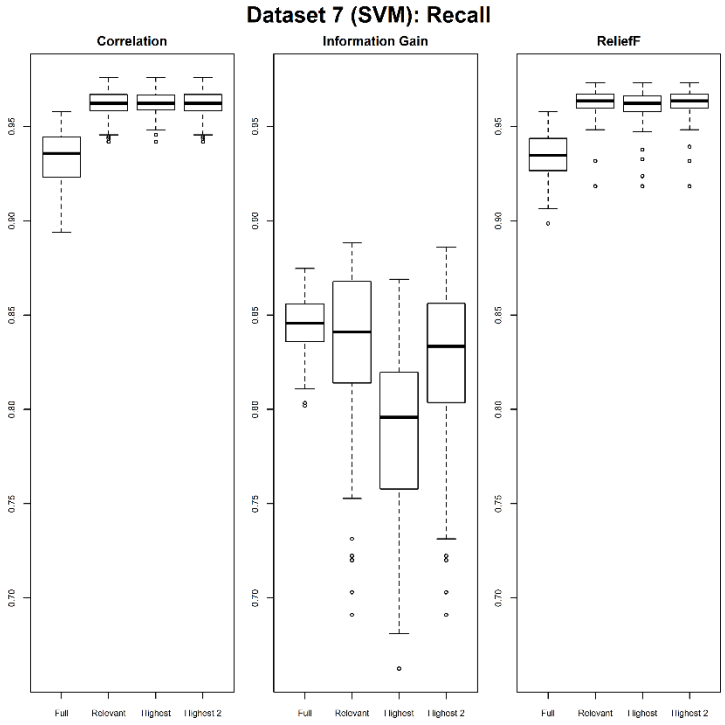


Figure B.28 Comparison of Recall using the SVM classifier: Dataset 7.

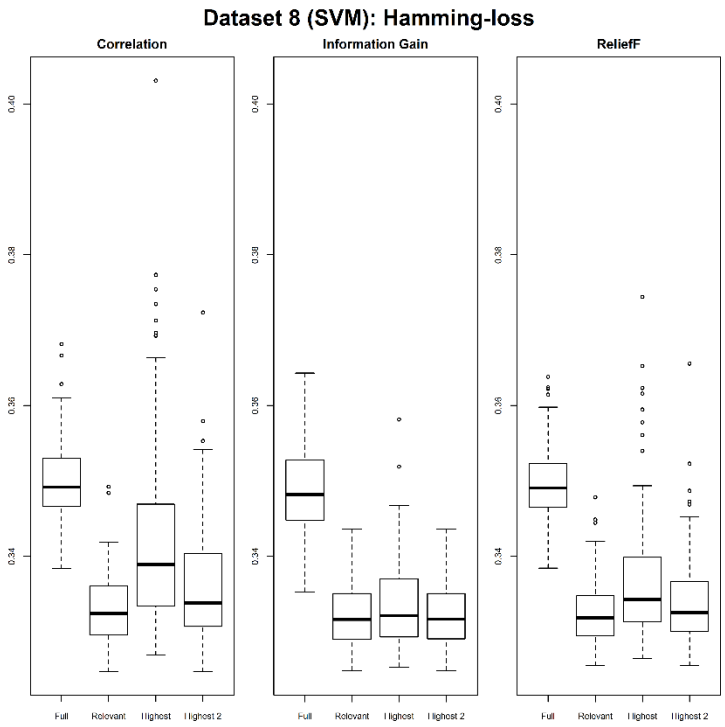


Figure B.29 Comparison of Hamming-loss using the SVM classifier: Dataset 8.

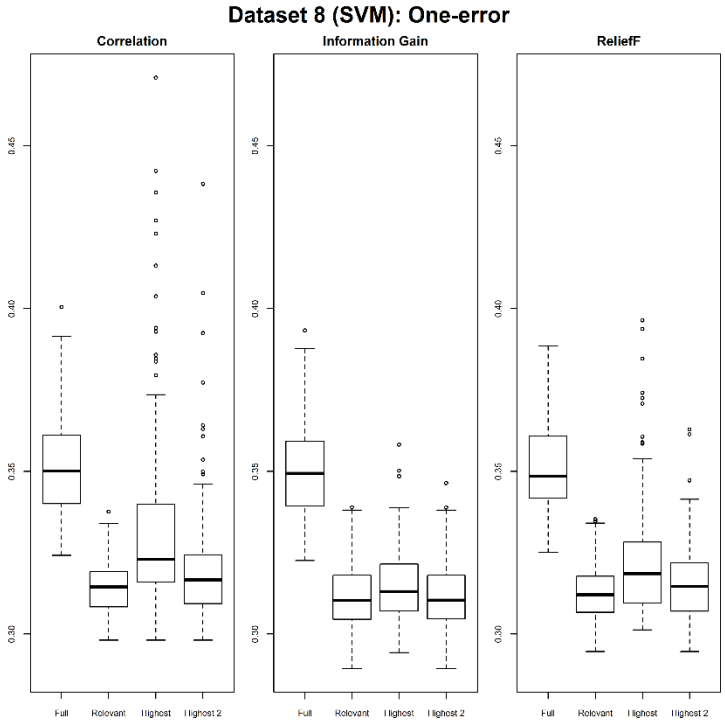


Figure B.30 Comparison of One-error using the SVM classifier: Dataset 8.

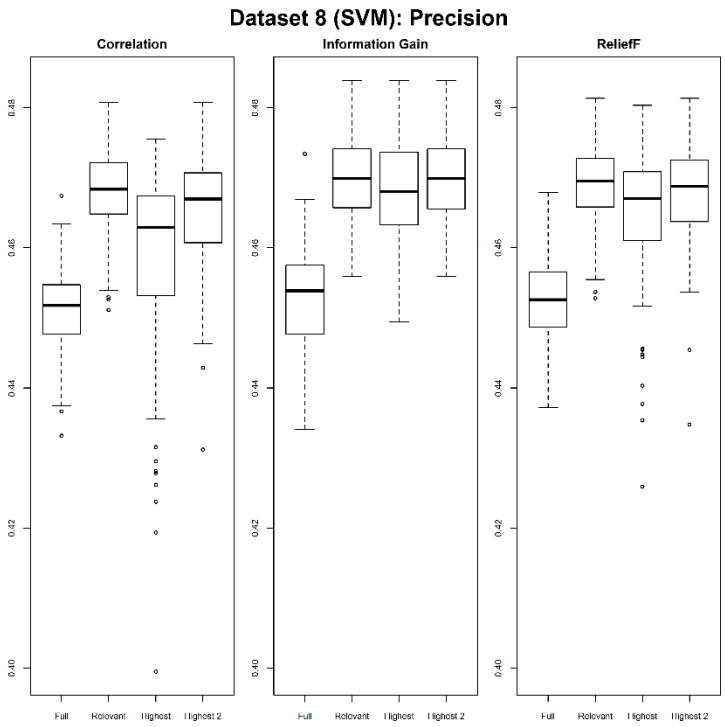


Figure B.31 Comparison of Precision using the SVM classifier: Dataset 8.

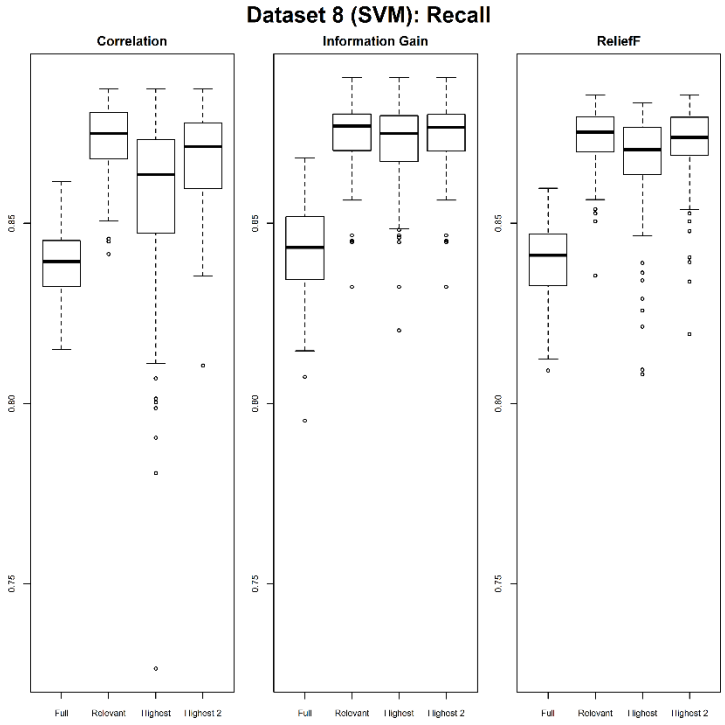


Figure B.32 Comparison of Recall using the SVM classifier: Dataset 8.

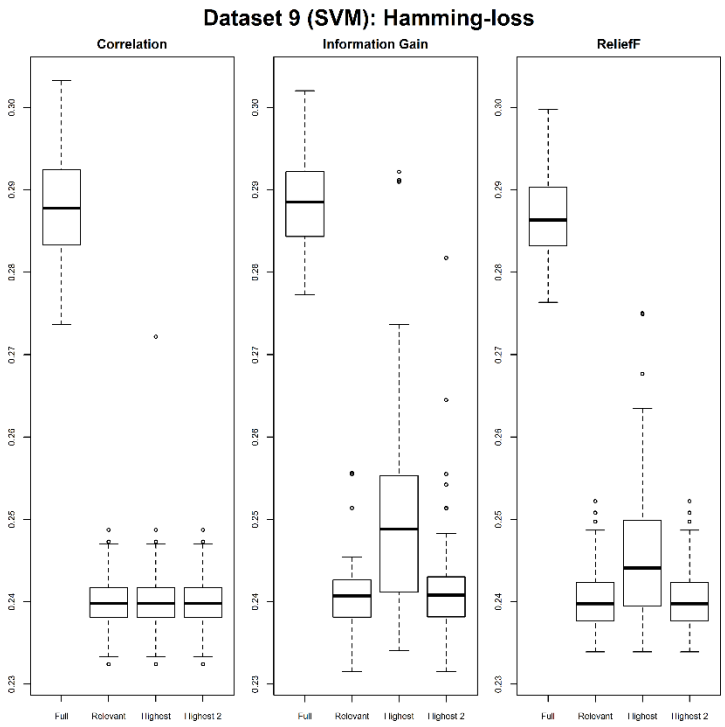


Figure B.33 Comparison of Hamming-loss using the SVM classifier: Dataset 9.

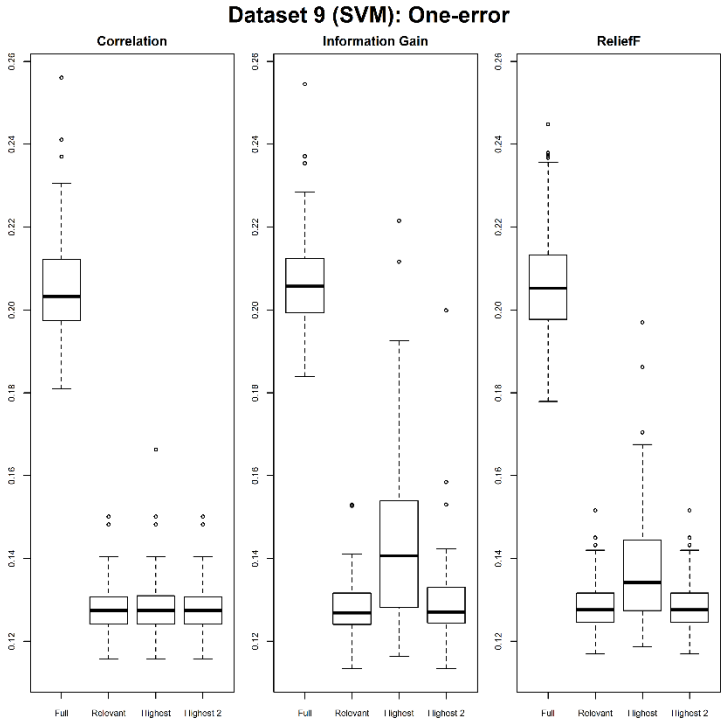


Figure B.34 Comparison of One-error using the SVM classifier: Dataset 9.

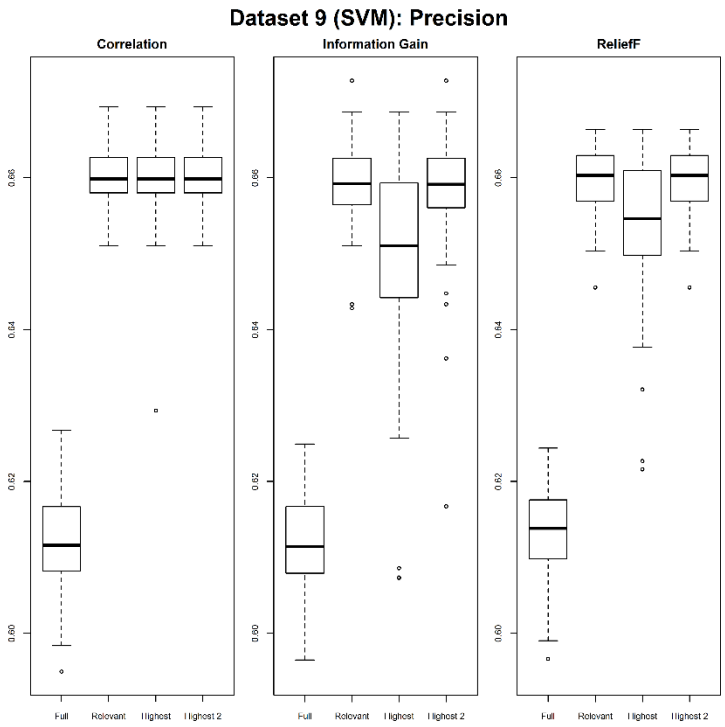


Figure B.35 Comparison of Precision using the SVM classifier: Dataset 9.

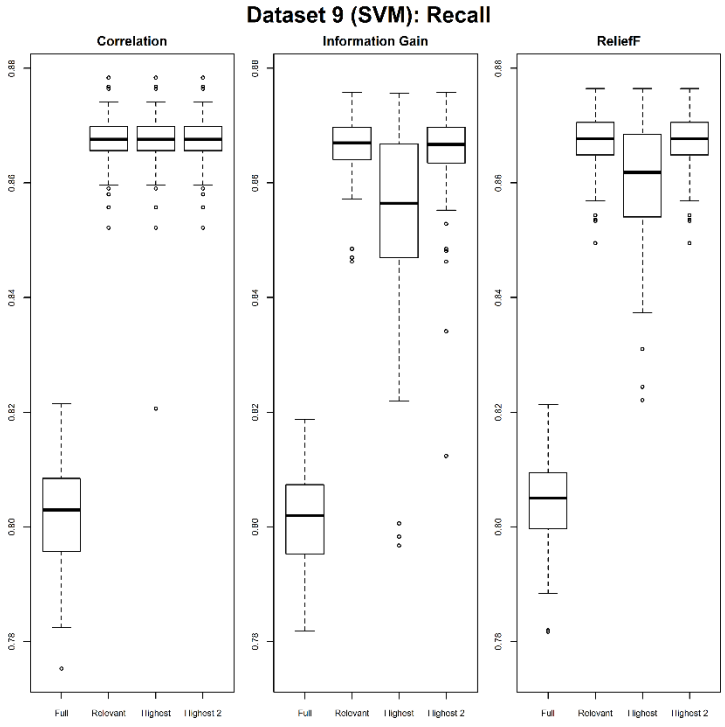


Figure B.36 Comparison of Recall using the SVM classifier: Dataset 9.

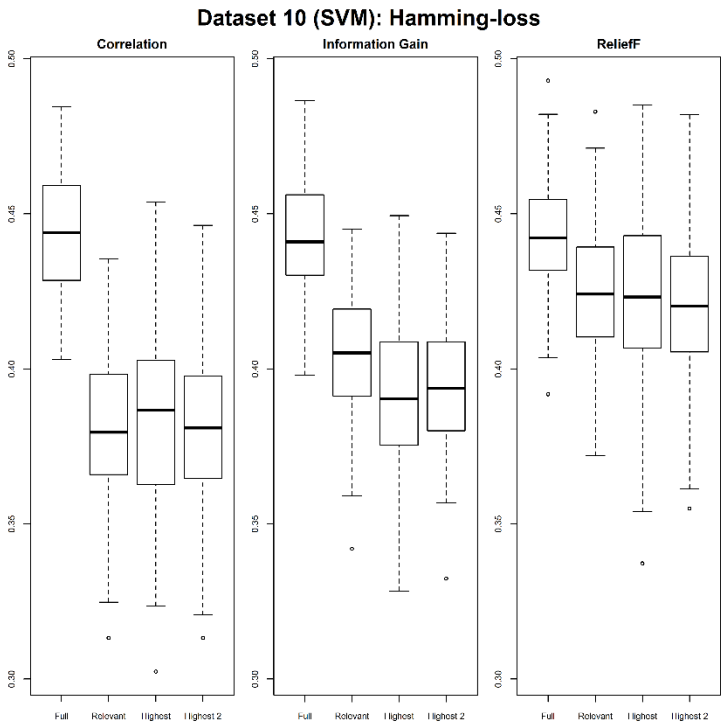


Figure B.37 Comparison of Hamming-loss using the SVM classifier: Dataset 10.

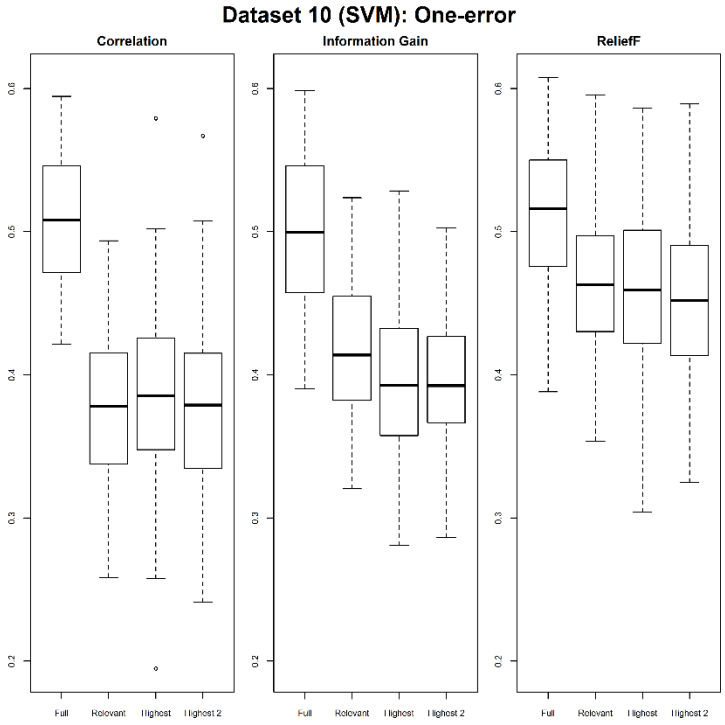


Figure B.38 Comparison of One-error using the SVM classifier: Dataset 10.

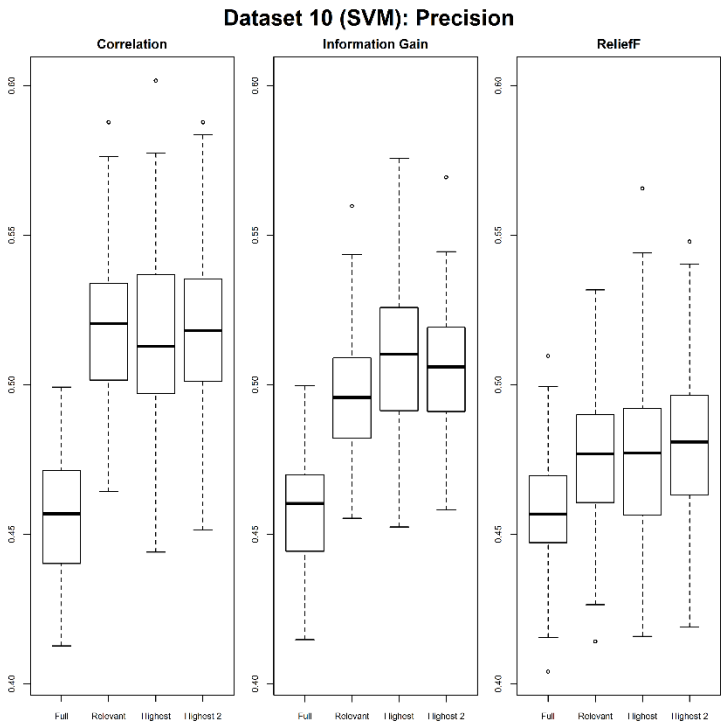


Figure B.39 Comparison of Precision using the SVM classifier: Dataset 10.

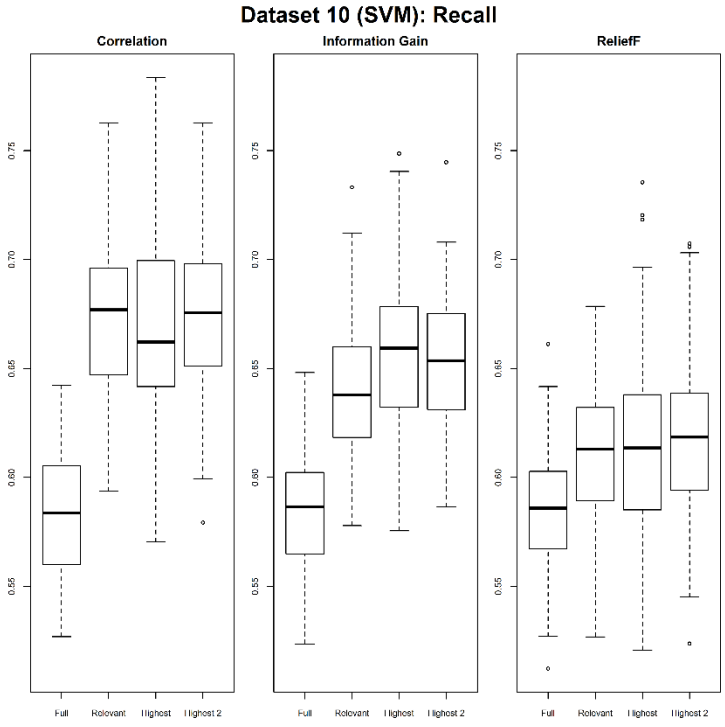


Figure B.40 Comparison of Recall using the SVM classifier: Dataset 10.

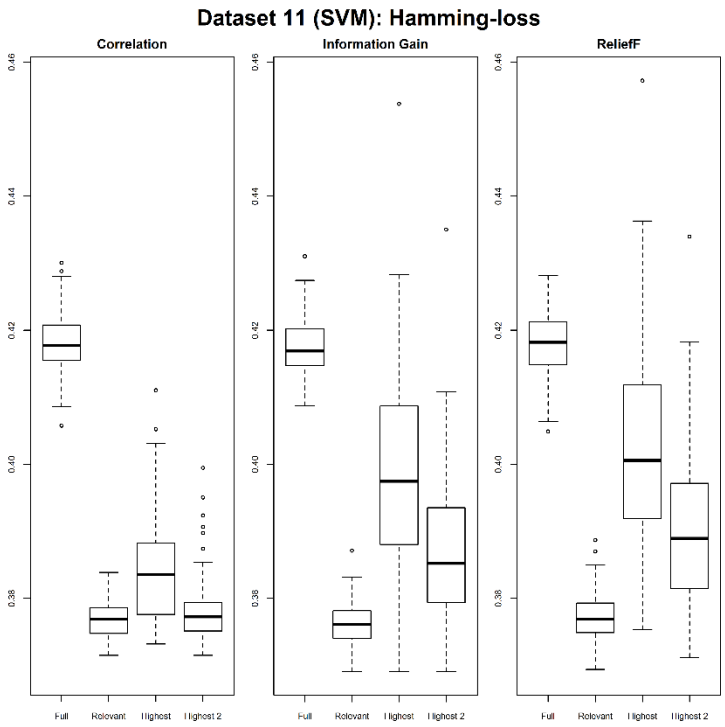


Figure B.41 Comparison of Hamming-loss using the SVM classifier: Dataset 11.

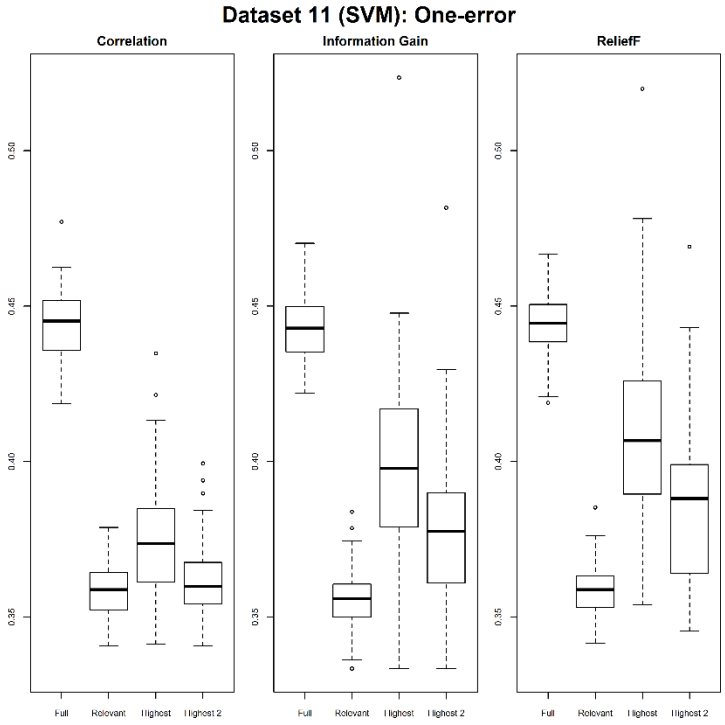


Figure B.42 Comparison of One-error using the SVM classifier: Dataset 11.

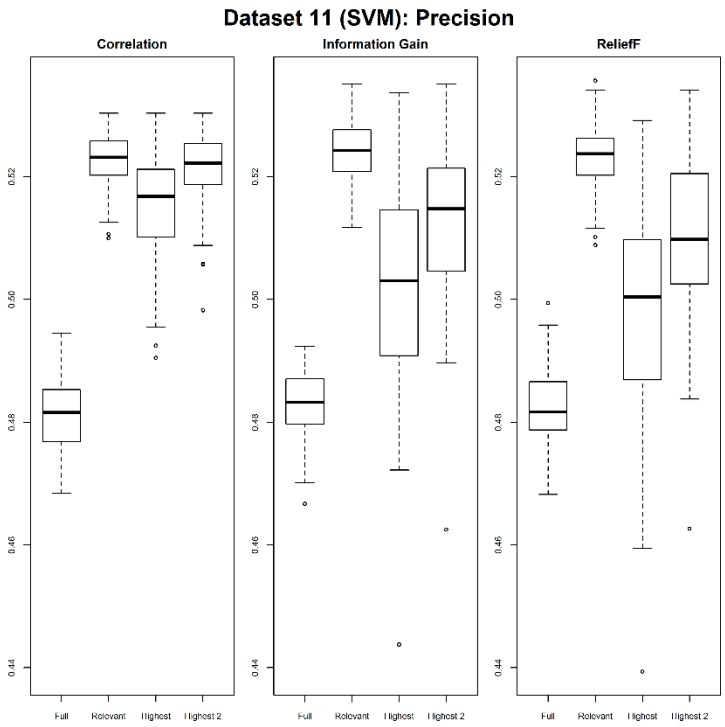


Figure B.43 Comparison of Precision using the SVM classifier: Dataset 11.

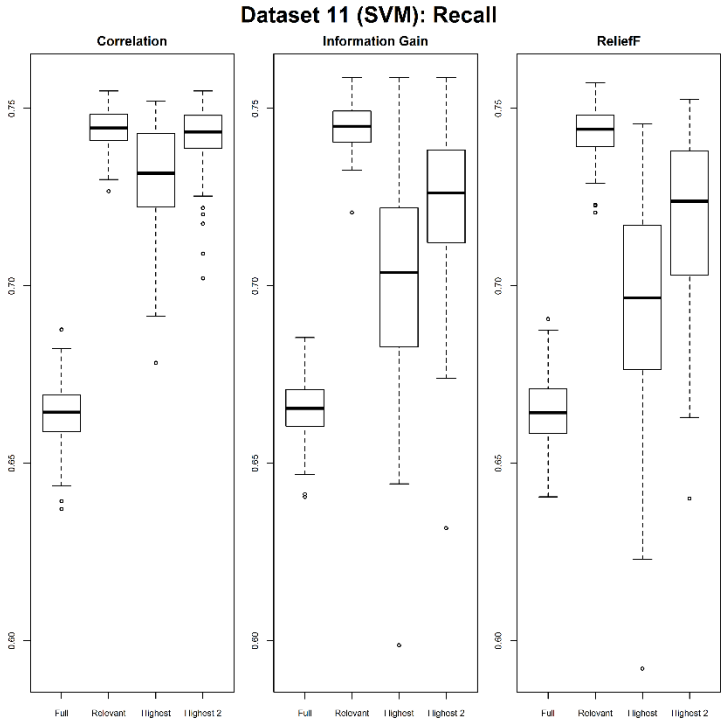


Figure B.44 Comparison of Recall using the SVM classifier: Dataset 11.

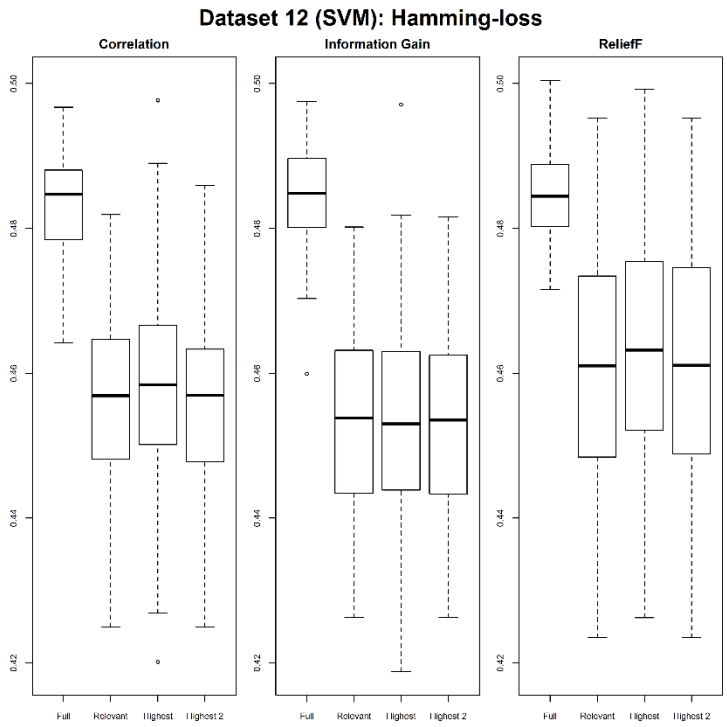


Figure B.45 Comparison of Hamming-loss using the SVM classifier: Dataset 12.

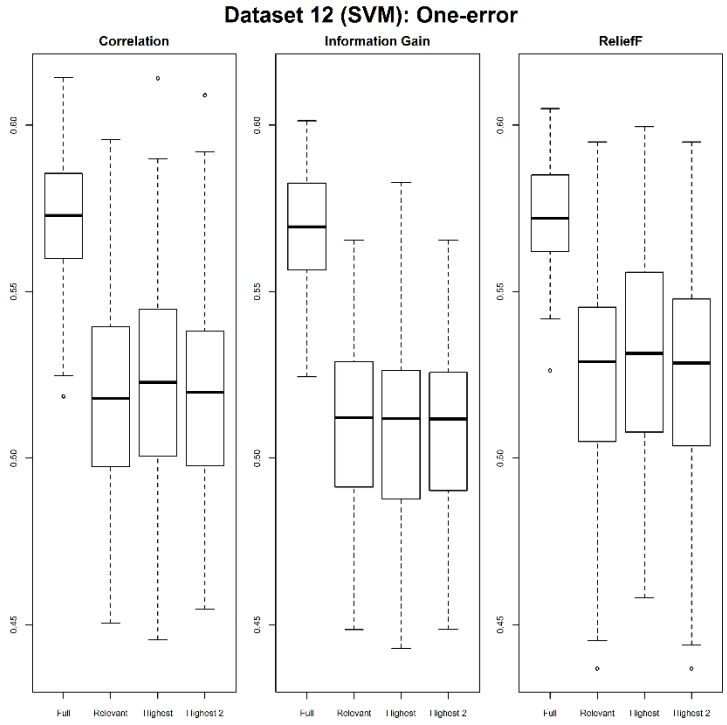


Figure B.46 Comparison of One-error using the SVM classifier: Dataset 12.

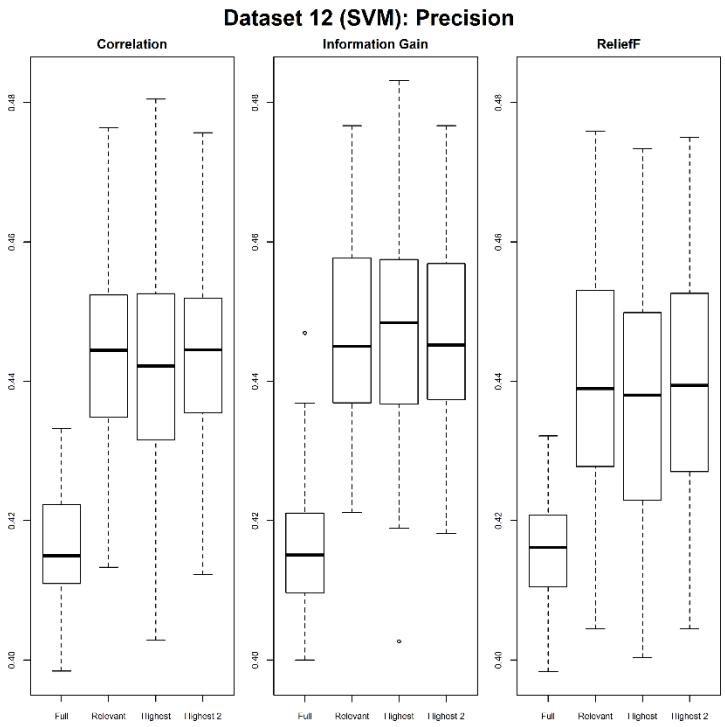


Figure B.47 Comparison of Precision using the SVM classifier: Dataset 12.

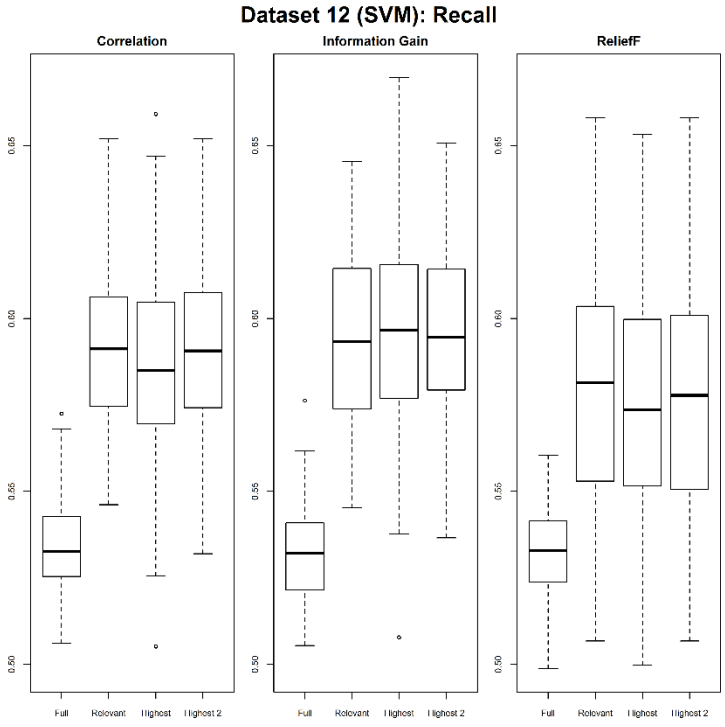


Figure B.48 Comparison of Recall using the SVM classifier: Dataset 12.

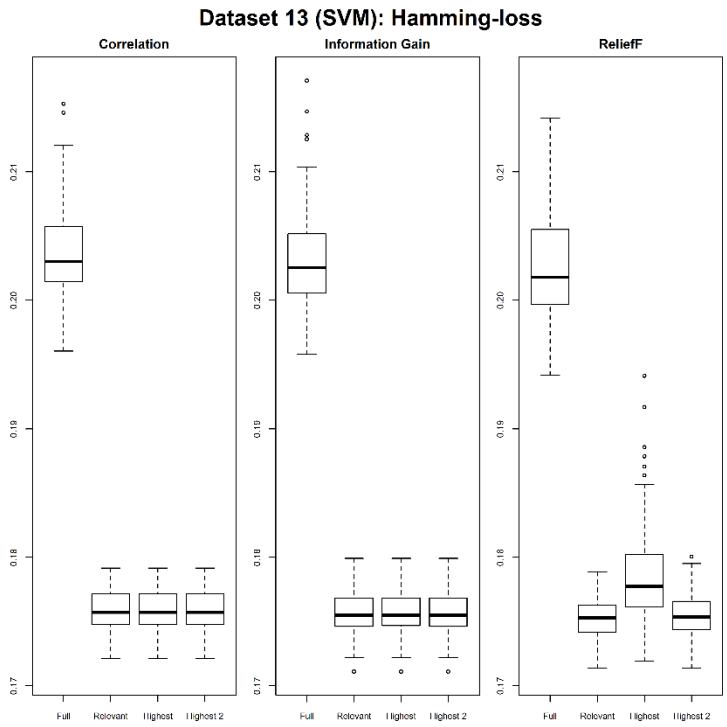


Figure B.49 Comparison of Hamming-loss using the SVM classifier: Dataset 13.

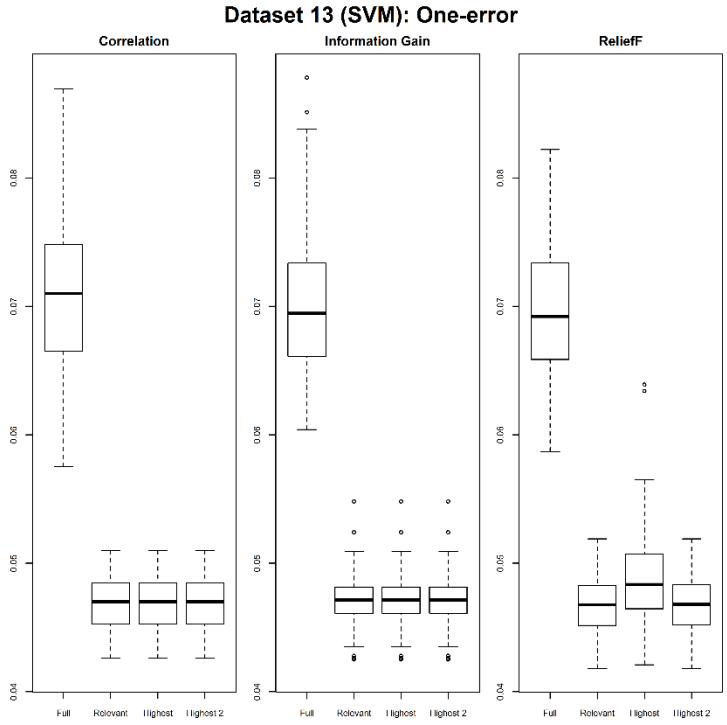


Figure B.50 Comparison of One-error using the SVM classifier: Dataset 13.

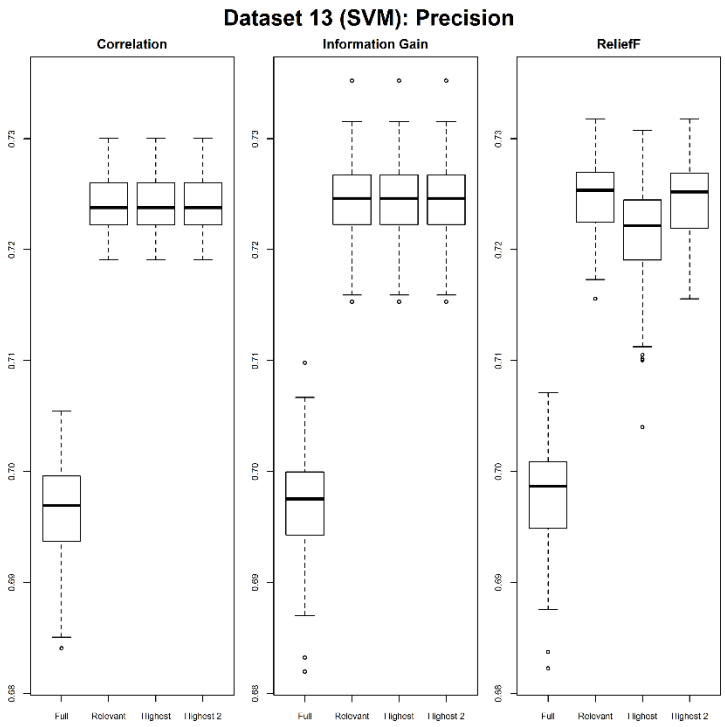


Figure B.51 Comparison of Precision using the SVM classifier: Dataset 13.

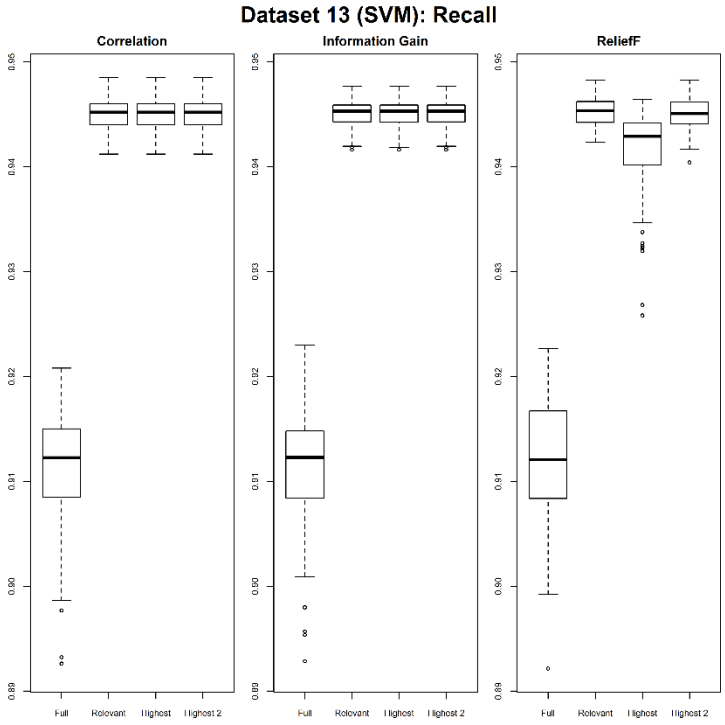


Figure B.52 Comparison of Recall using the SVM classifier: Dataset 13.

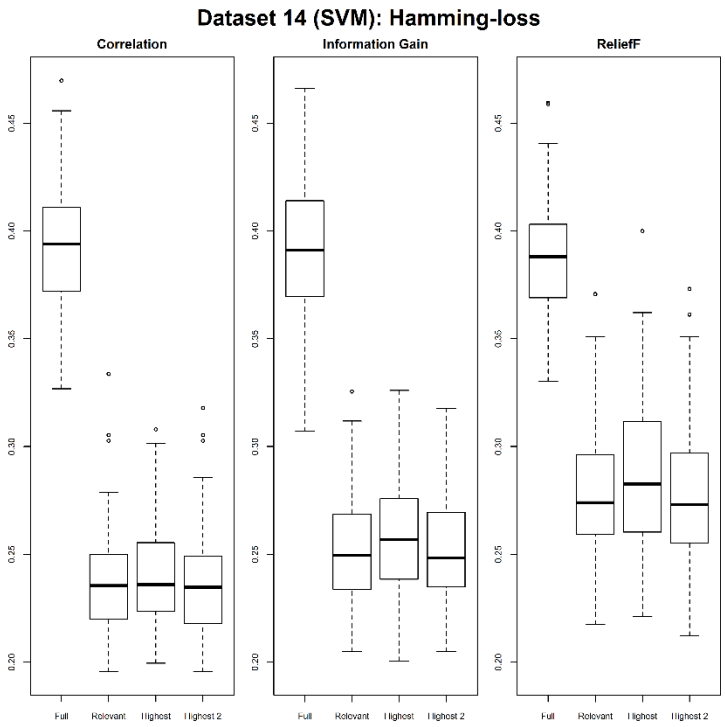


Figure B.53 Comparison of Hamming-loss using the SVM classifier: Dataset 14.

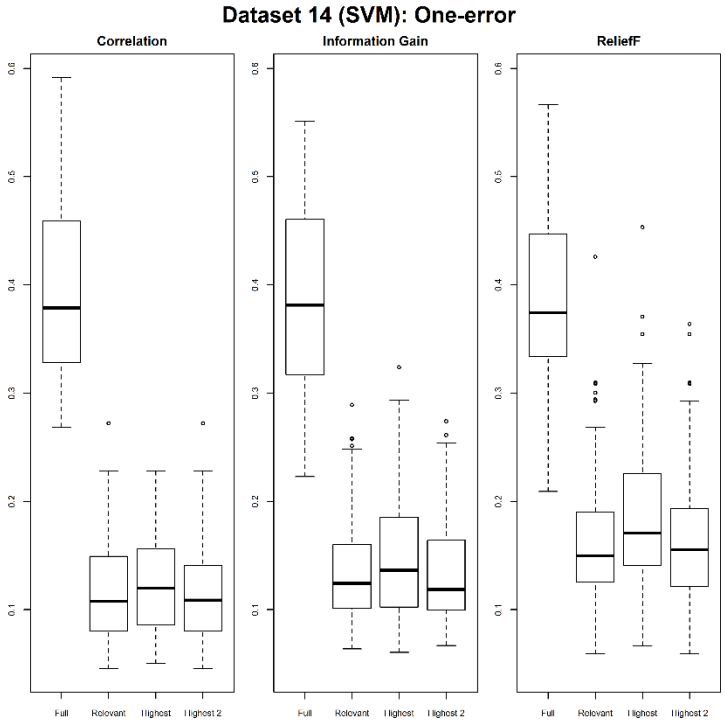


Figure B.54 Comparison of One-error using the SVM classifier: Dataset 14.

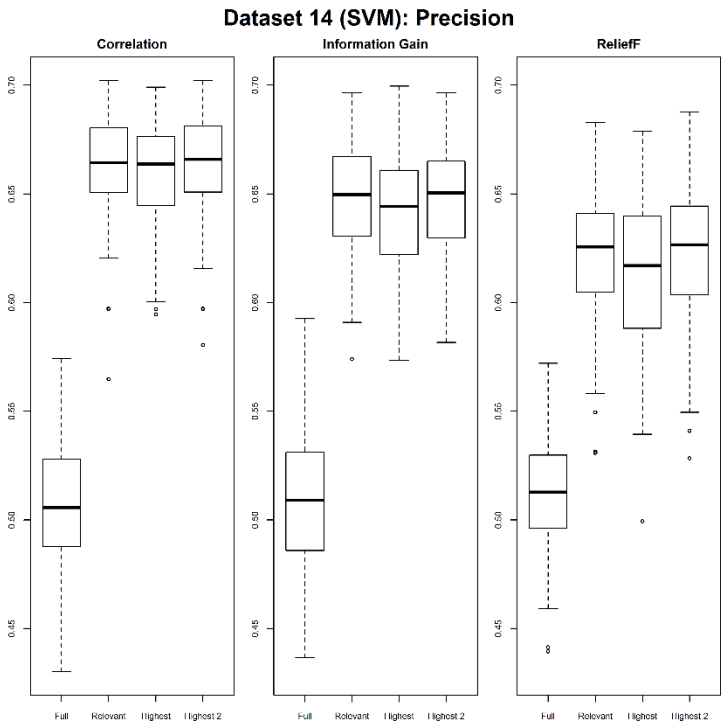


Figure B.55 Comparison of Precision using the SVM classifier: Dataset 14.

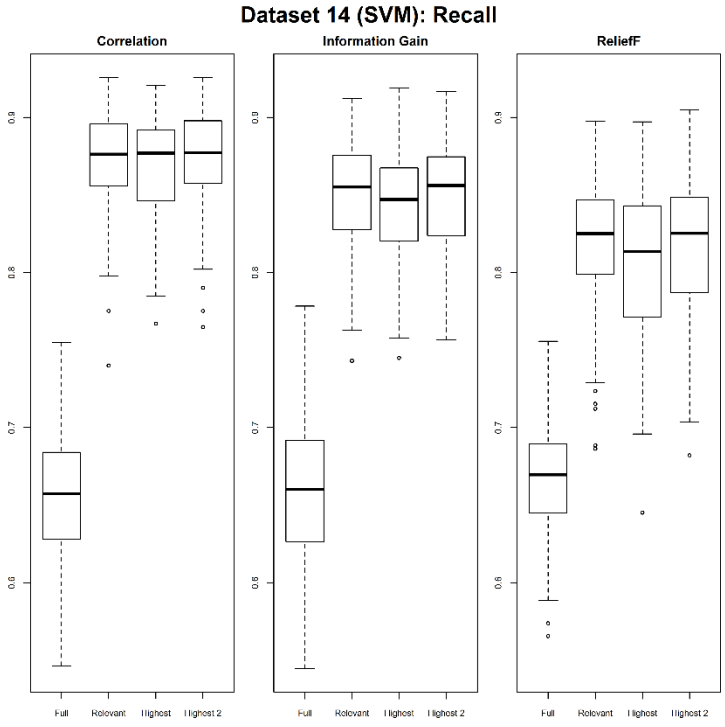


Figure B.56 Comparison of Recall using the SVM classifier: Dataset 14.

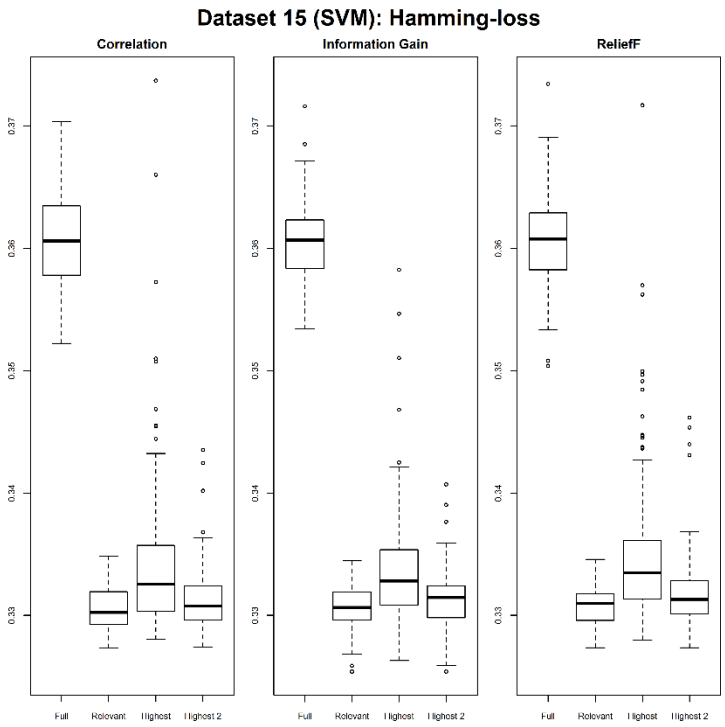


Figure B.57 Comparison of Hamming-loss using the SVM classifier: Dataset 15.

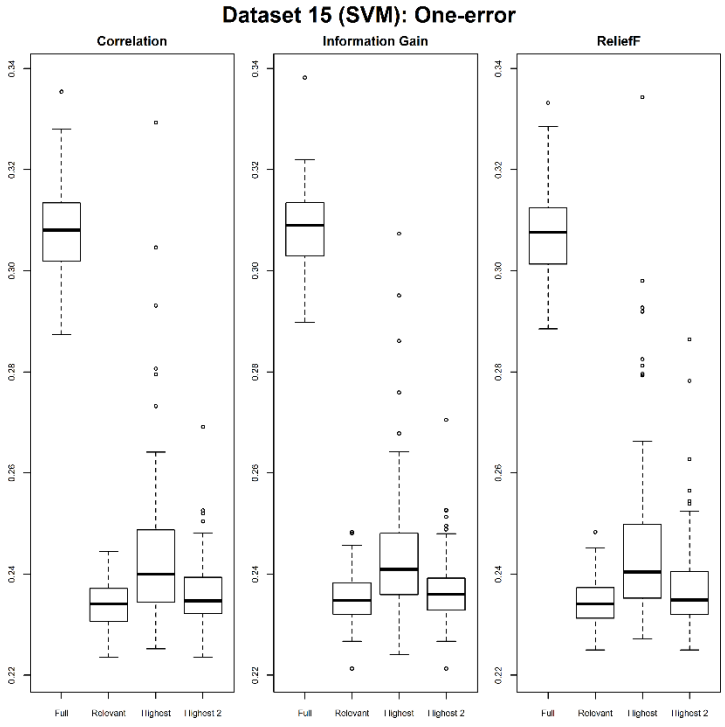


Figure B.58 Comparison of One-error using the SVM classifier: Dataset 15.

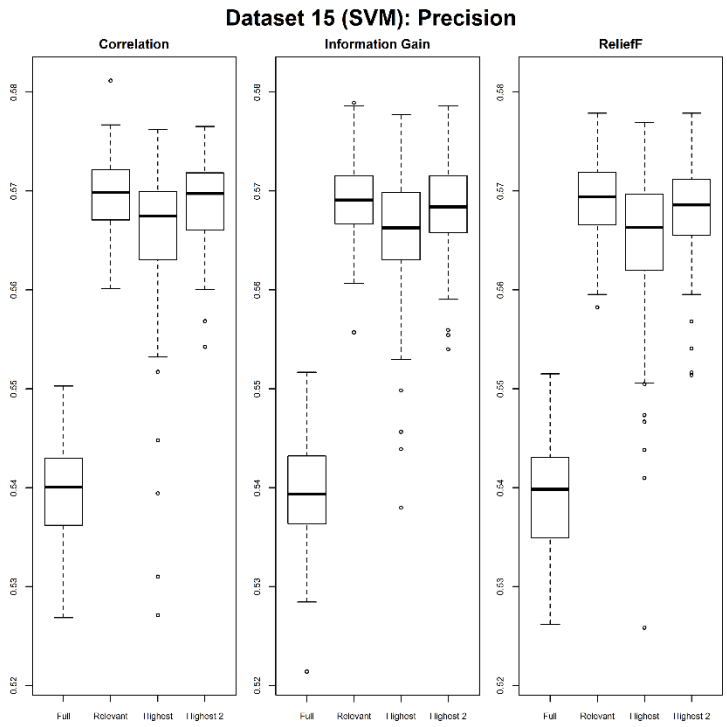


Figure B.59 Comparison of Precision using the SVM classifier: Dataset 15.

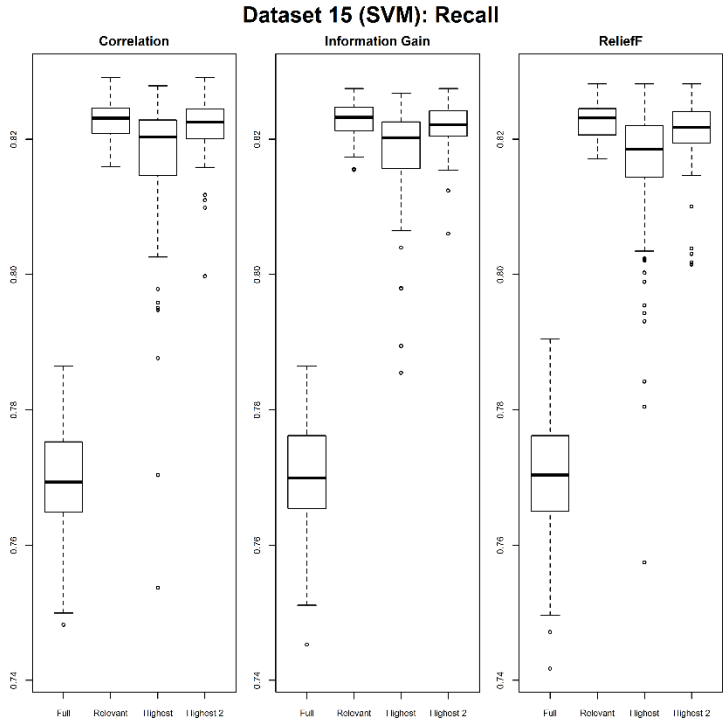


Figure B.60 Comparison of Recall using the SVM classifier: Dataset 15.

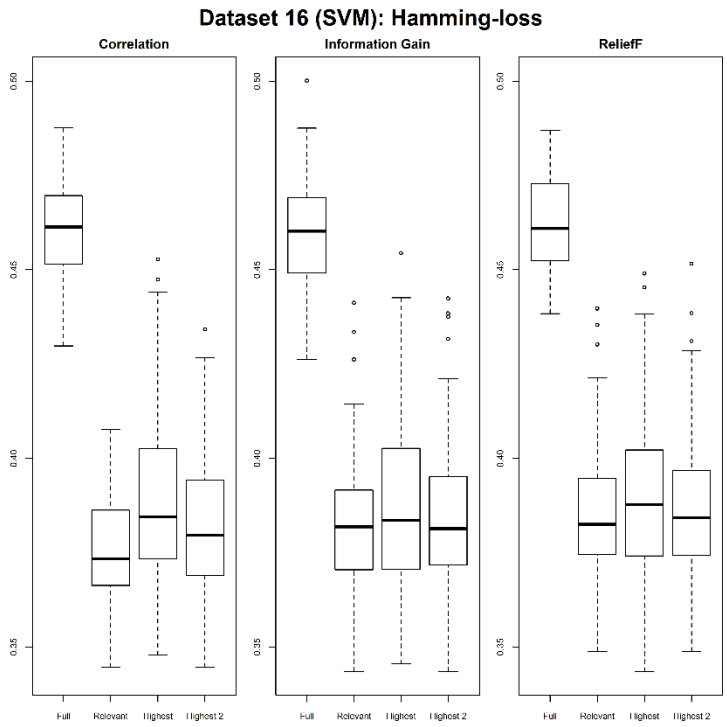


Figure B.61 Comparison of Hamming-loss using the SVM classifier: Dataset 16.

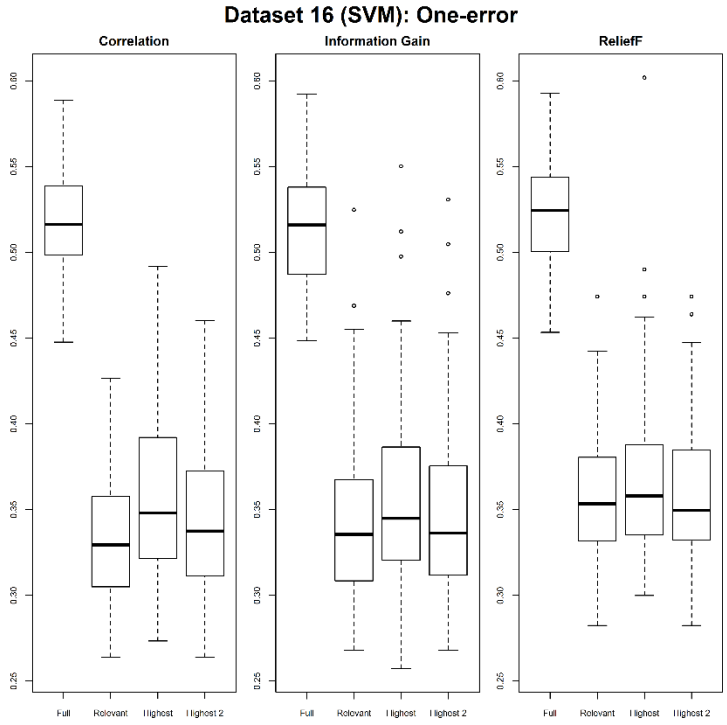


Figure B.62 Comparison of One-error using the SVM classifier: Dataset 16.

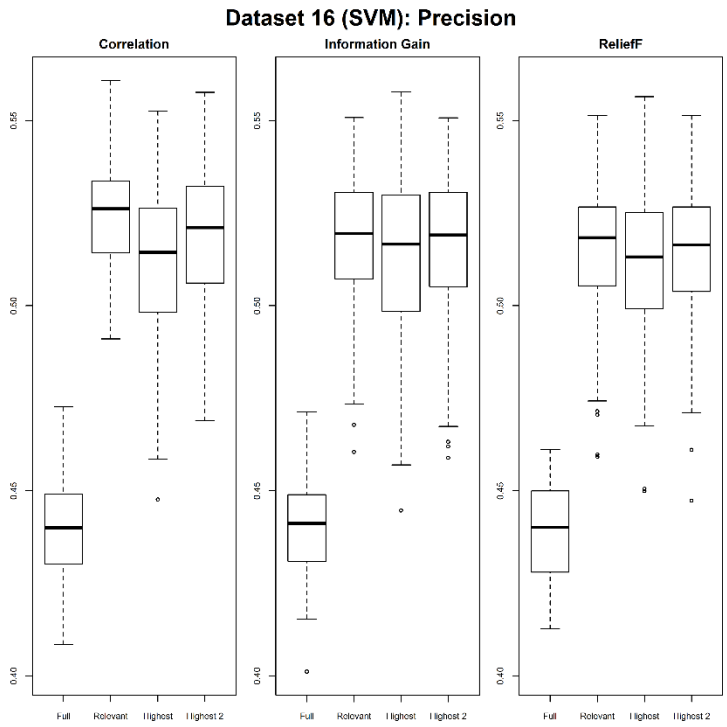


Figure B.63 Comparison of Precision using the SVM classifier: Dataset 16.

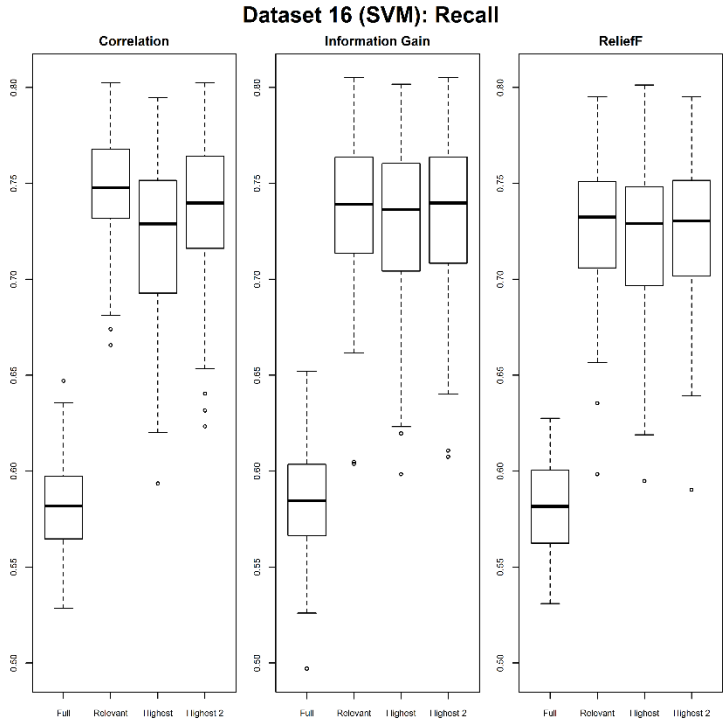


Figure B.64 Comparison of Recall using the SVM classifier: Dataset 16.

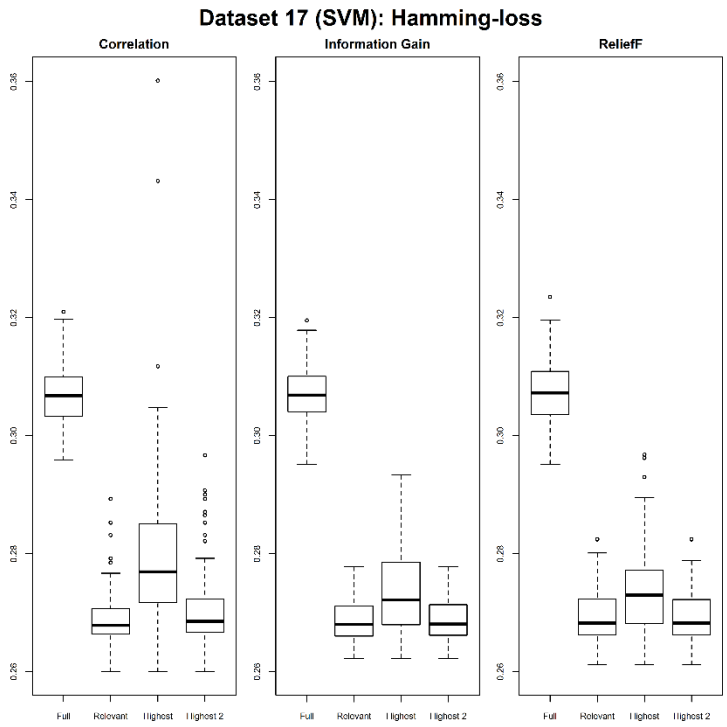


Figure B.65 Comparison of Hamming-loss using the SVM classifier: Dataset 17.

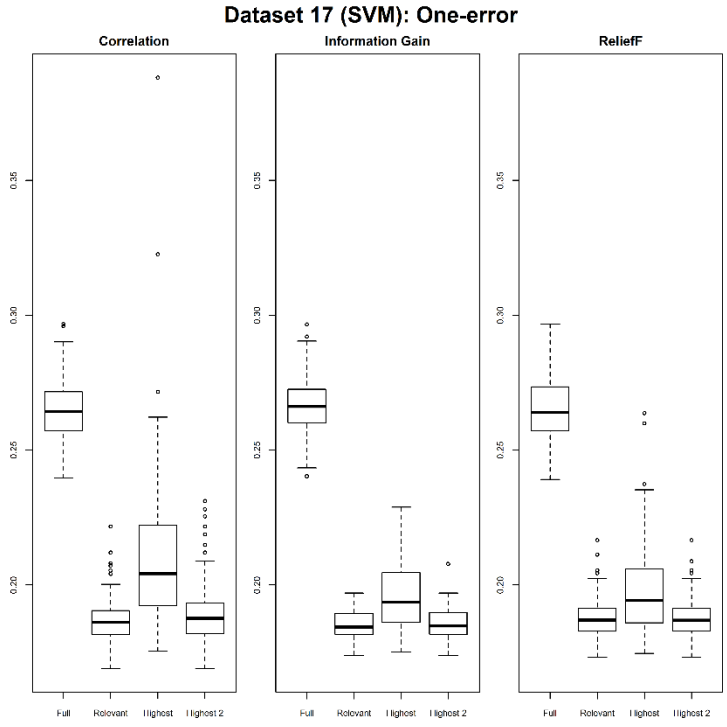


Figure B.66 Comparison of One-error using the SVM classifier: Dataset 17.

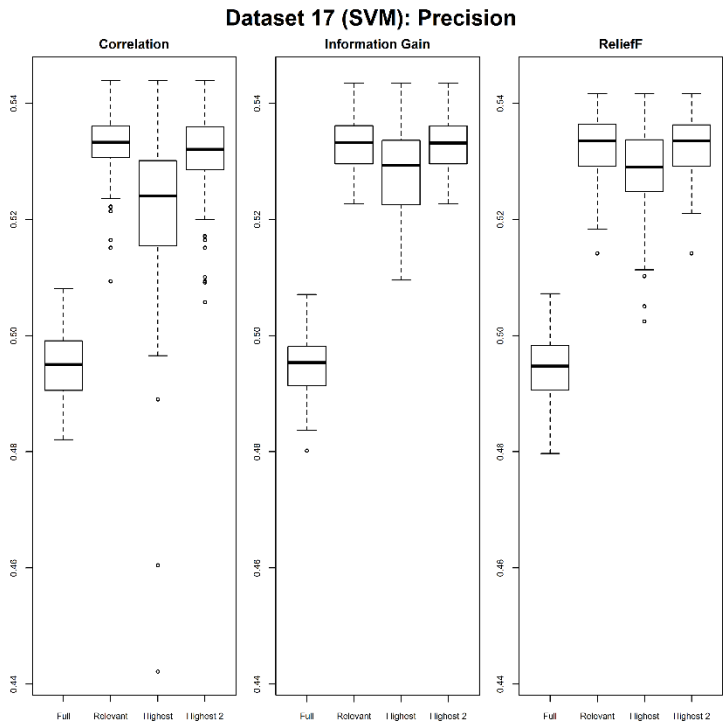


Figure B.67 Comparison of Precision using the SVM classifier: Dataset 17.

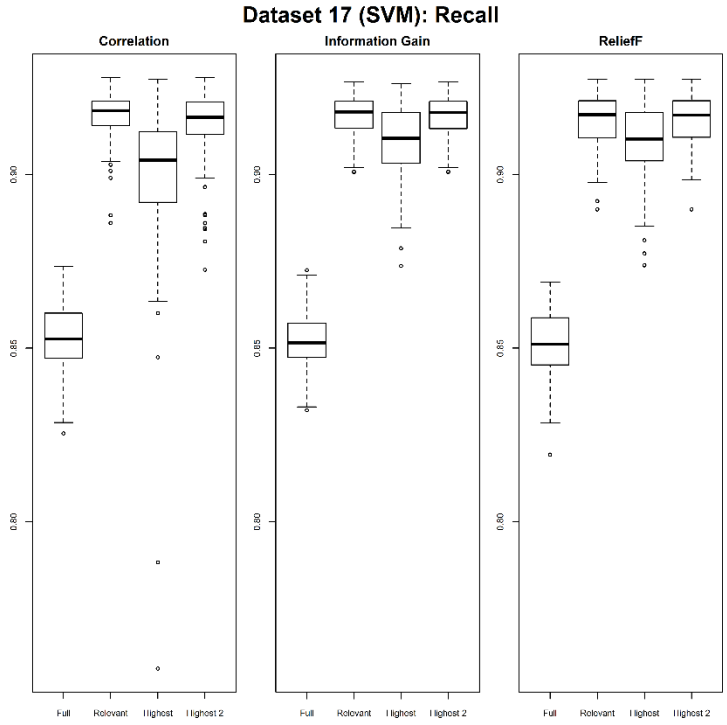


Figure B.68 Comparison of Recall using the SVM classifier: Dataset 17.

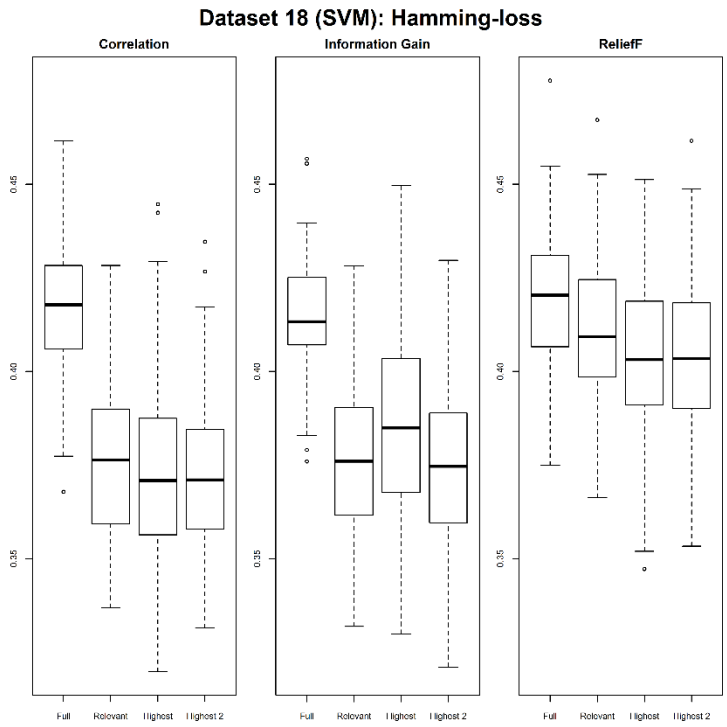


Figure B.69 Comparison of Hamming-loss using the SVM classifier: Dataset 18.

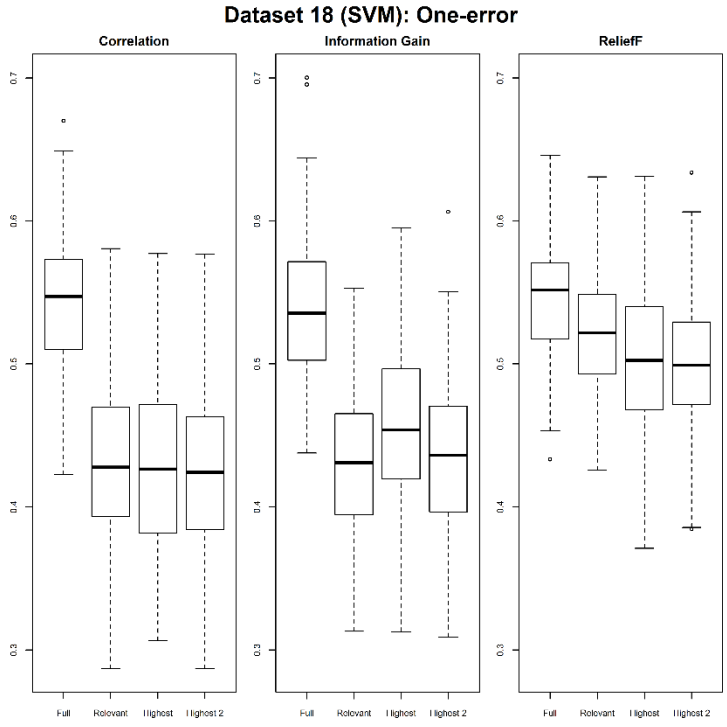


Figure B.70 Comparison of One-error using the SVM classifier: Dataset 18.

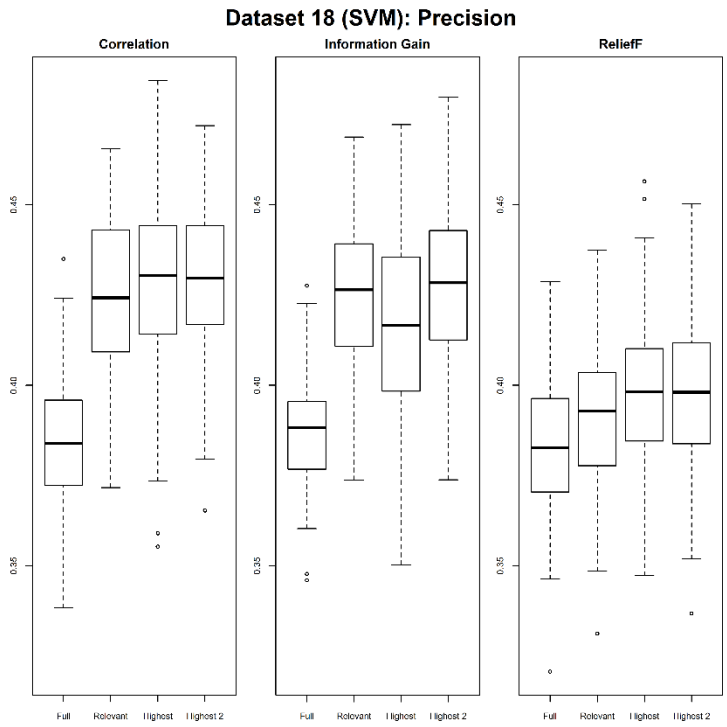


Figure B.71 Comparison of Precision using the SVM classifier: Dataset 18.

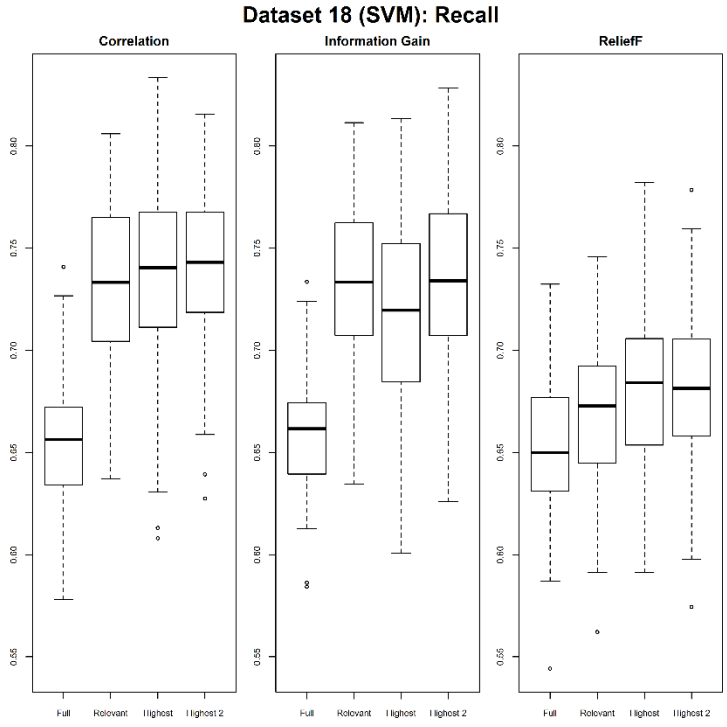


Figure B.72 Comparison of Recall using the SVM classifier: Dataset 18.

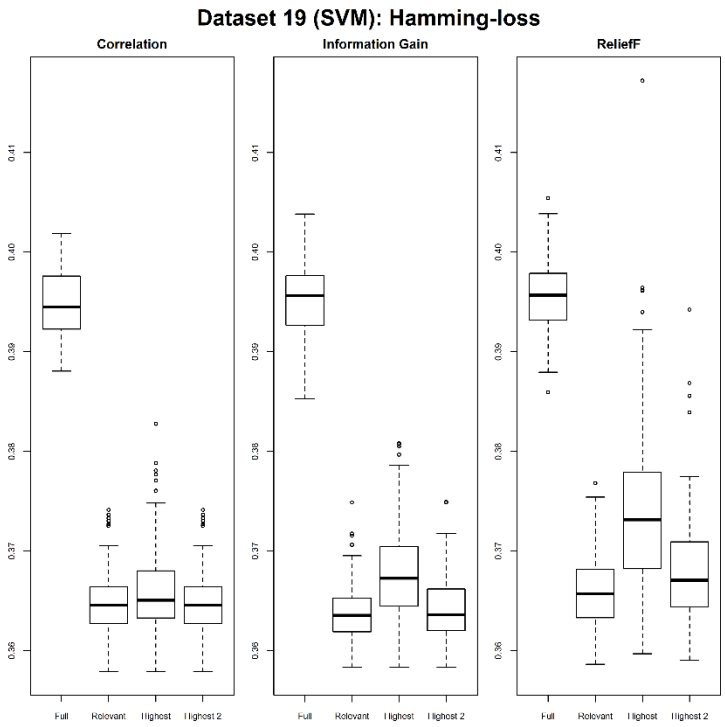


Figure B.73 Comparison of Hamming-loss using the SVM classifier: Dataset 19.

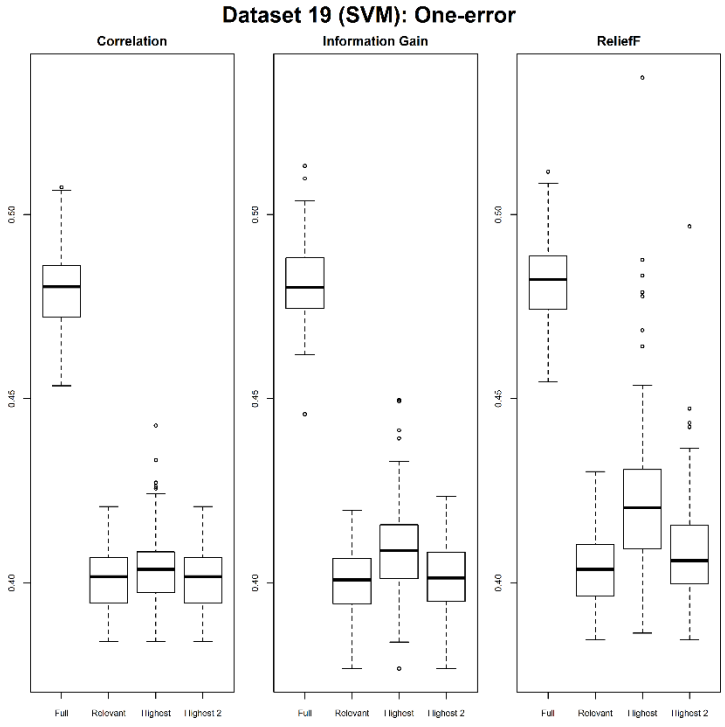


Figure B.74 Comparison of One-error using the SVM classifier: Dataset 19.

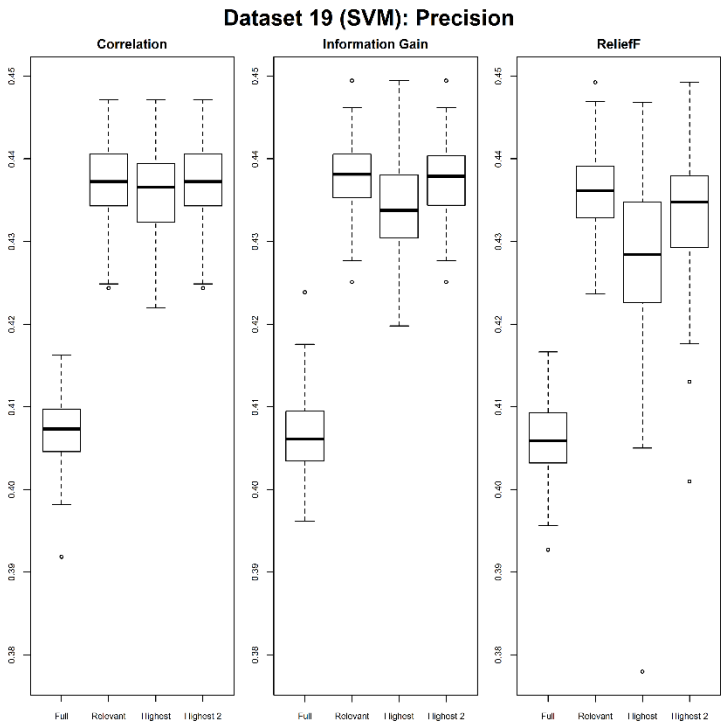


Figure B.75 Comparison of Precision using the SVM classifier: Dataset 19.

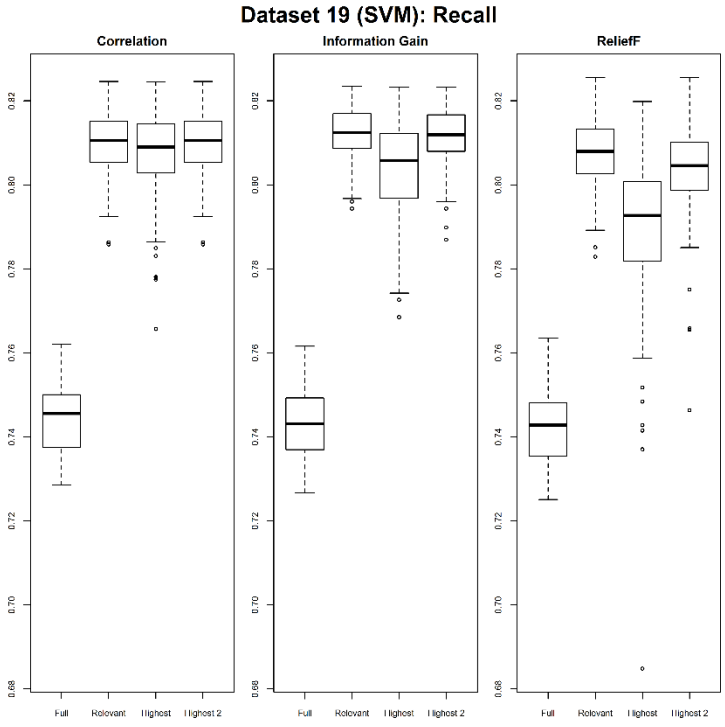


Figure B.76 Comparison of Recall using the SVM classifier: Dataset 19.

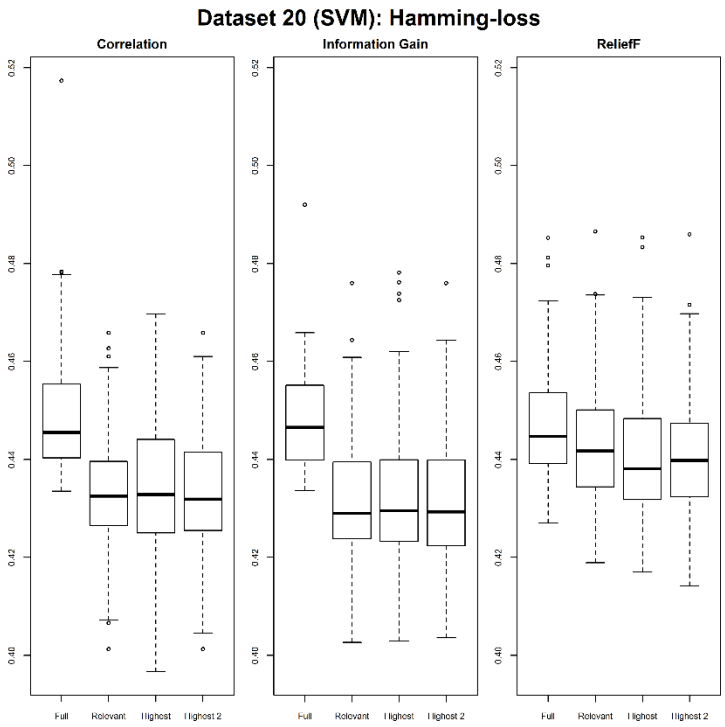


Figure B.77 Comparison of Hamming-loss using the SVM classifier: Dataset 20.

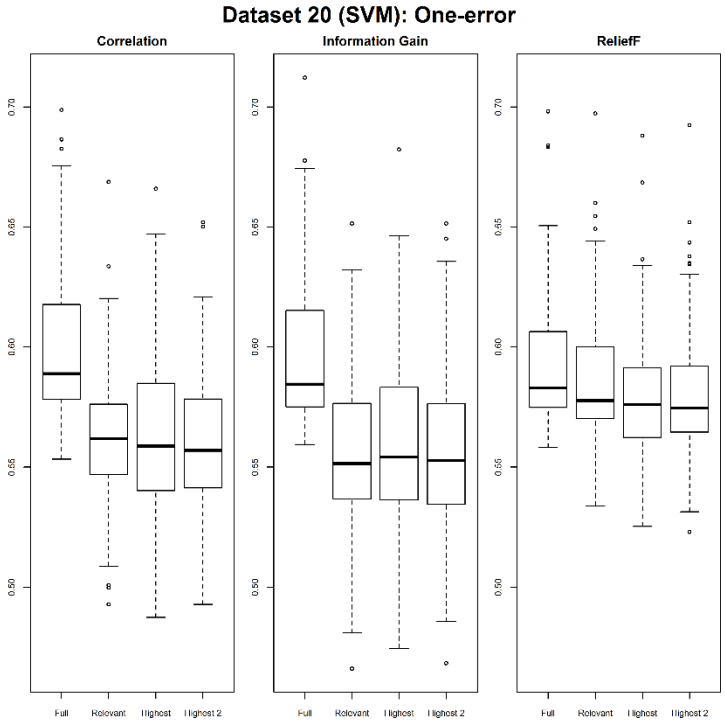


Figure B.78 Comparison of One-error using the SVM classifier: Dataset 20.

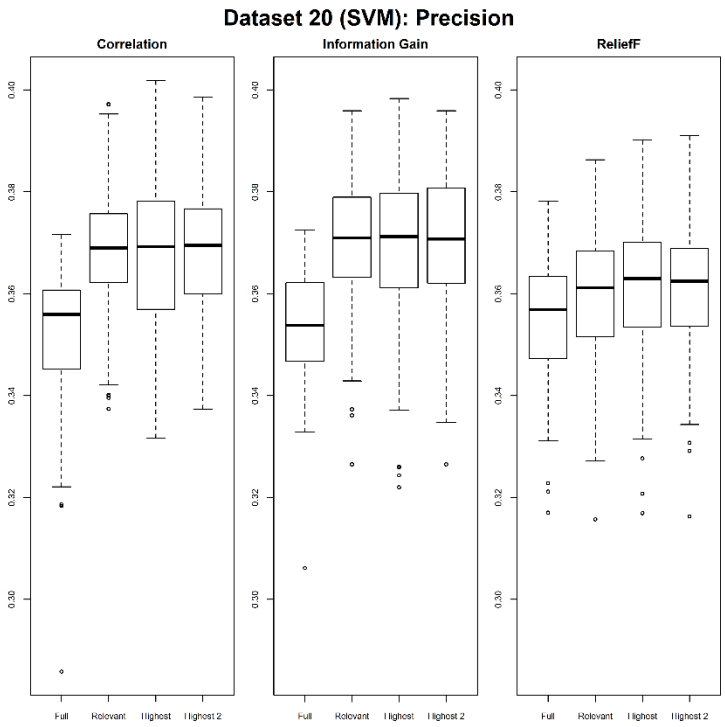


Figure B.79 Comparison of Precision using the SVM classifier: Dataset 20.

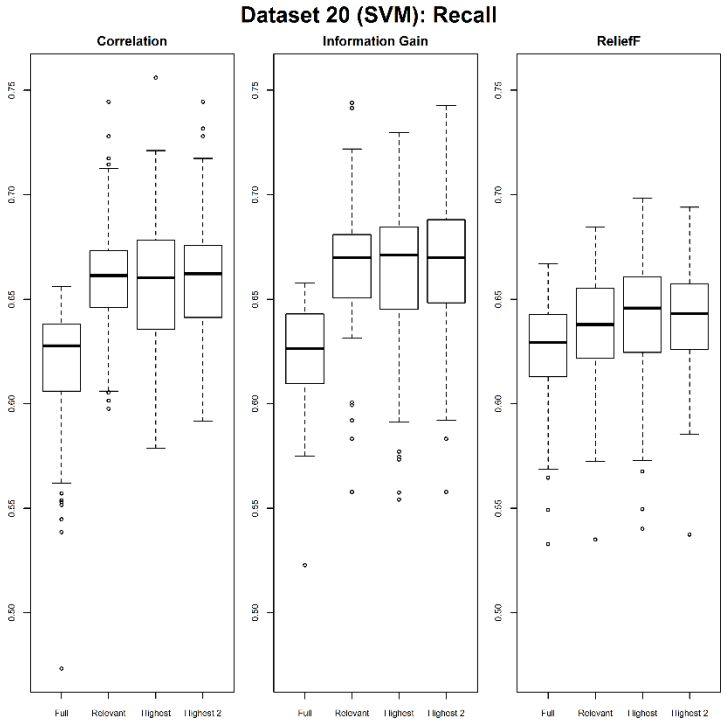


Figure B.80 Comparison of Recall using the SVM classifier: Dataset 20.

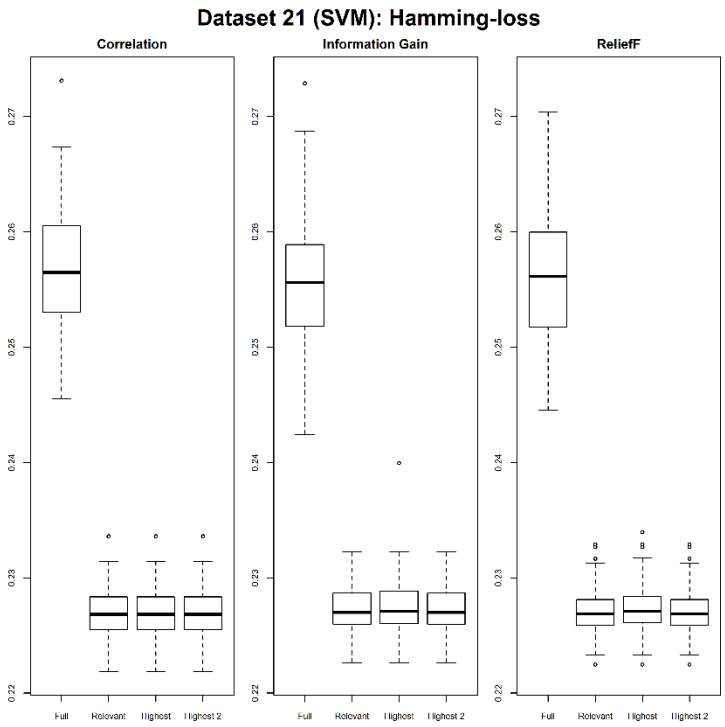


Figure B.81 Comparison of Hamming-loss using the SVM classifier: Dataset 21.

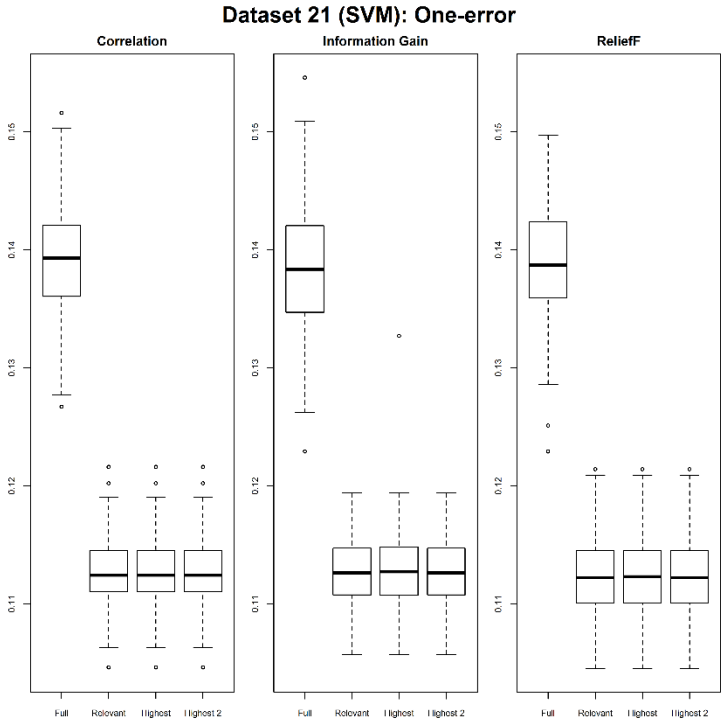


Figure B.82 Comparison of One-error using the SVM classifier: Dataset 21.

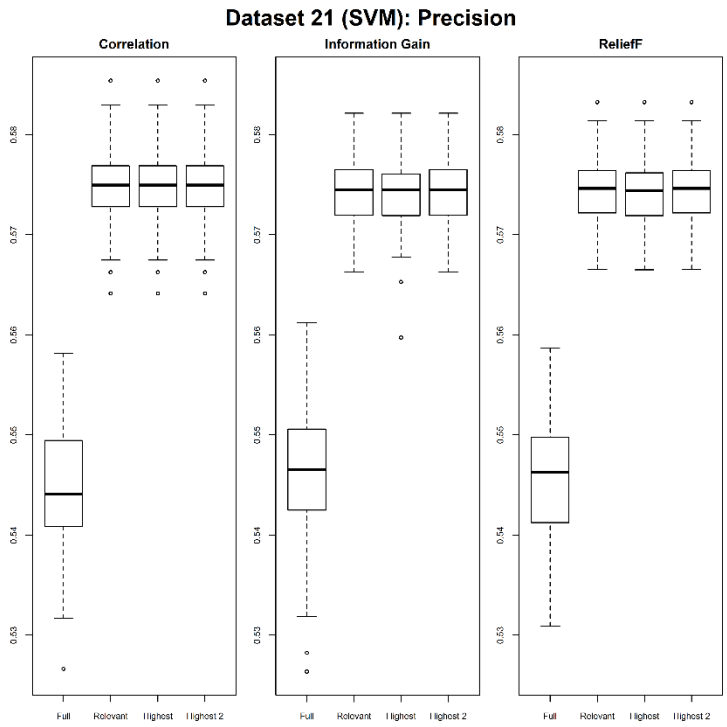


Figure B.83 Comparison of Precision using the SVM classifier: Dataset 21.

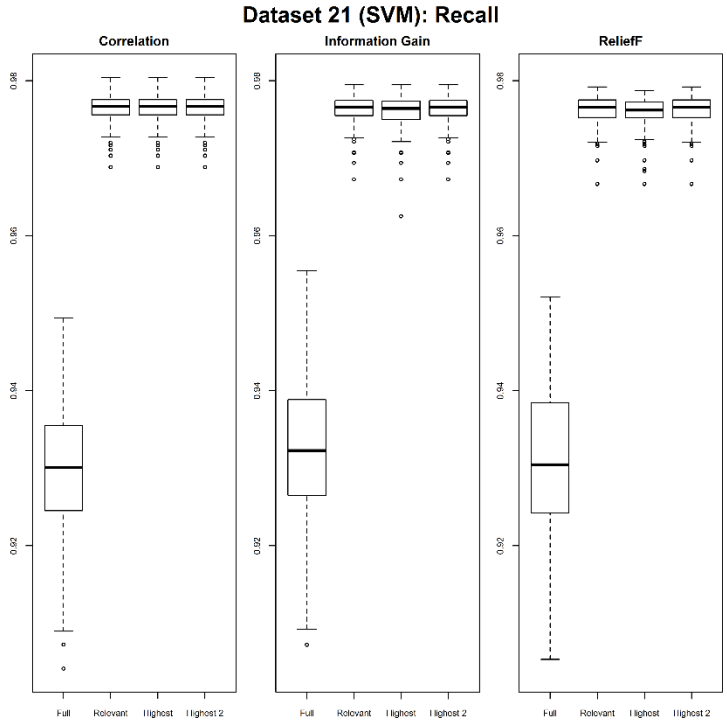


Figure B.84 Comparison of Recall using the SVM classifier: Dataset 21.

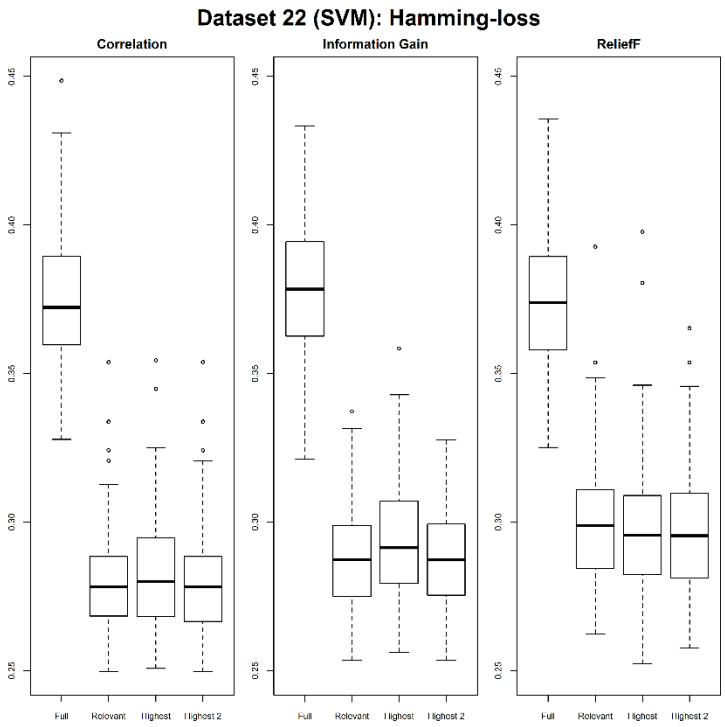


Figure B.85 Comparison of Hamming-loss using the SVM classifier: Dataset 22.

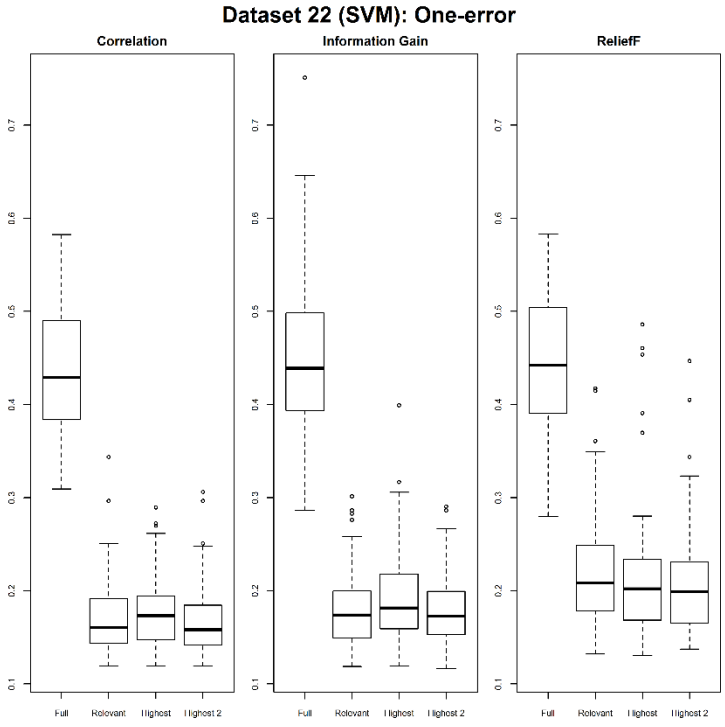


Figure B.86 Comparison of One-error using the SVM classifier: Dataset 22.

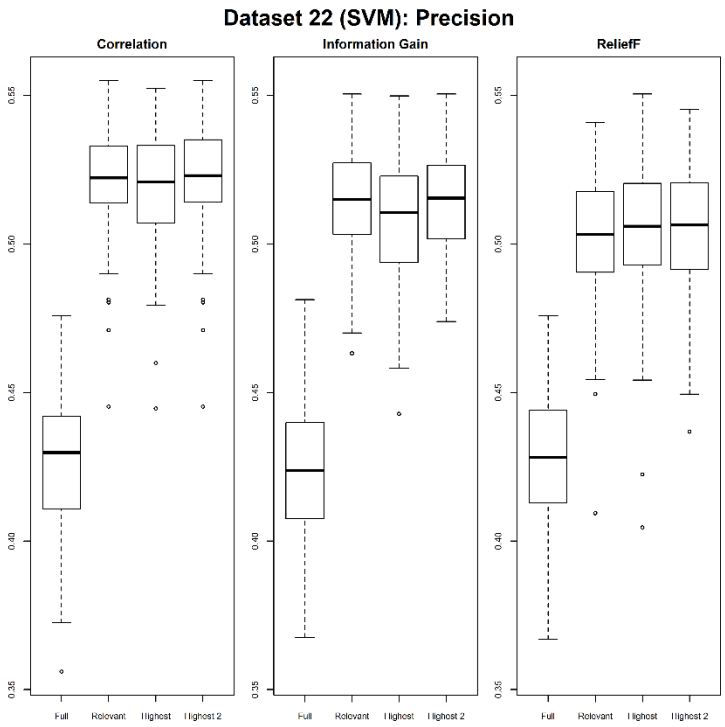


Figure B.87 Comparison of Precision using the SVM classifier: Dataset 22.

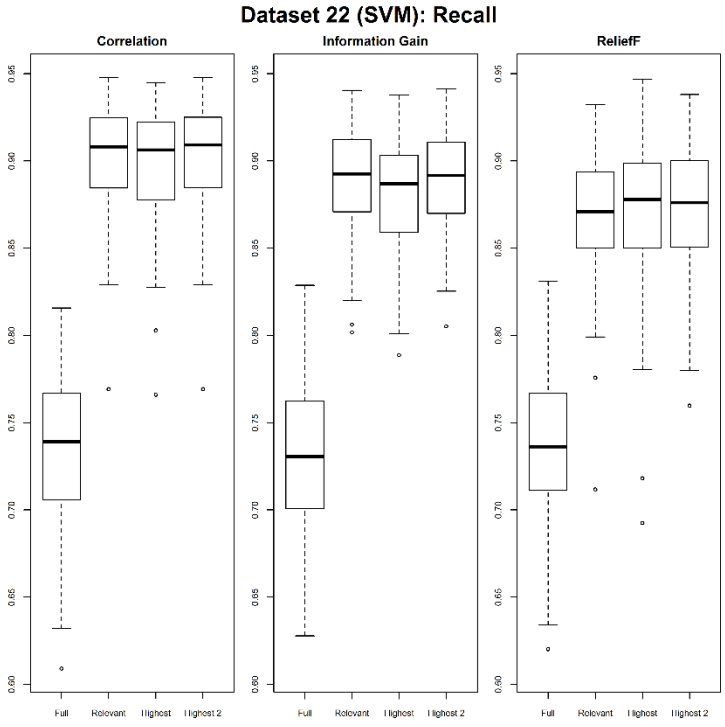


Figure B.88 Comparison of Recall using the SVM classifier: Dataset 22.

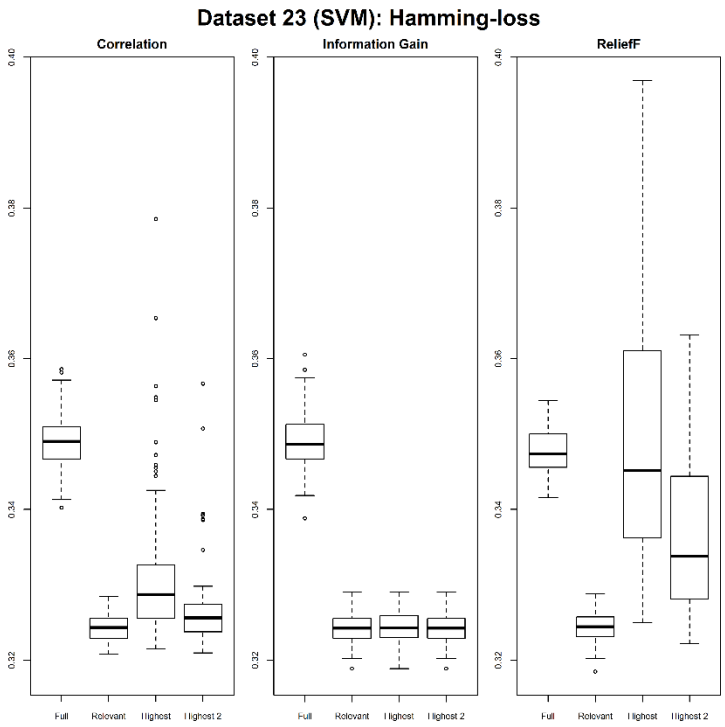


Figure B.89 Comparison of Hamming-loss using the SVM classifier: Dataset 23.

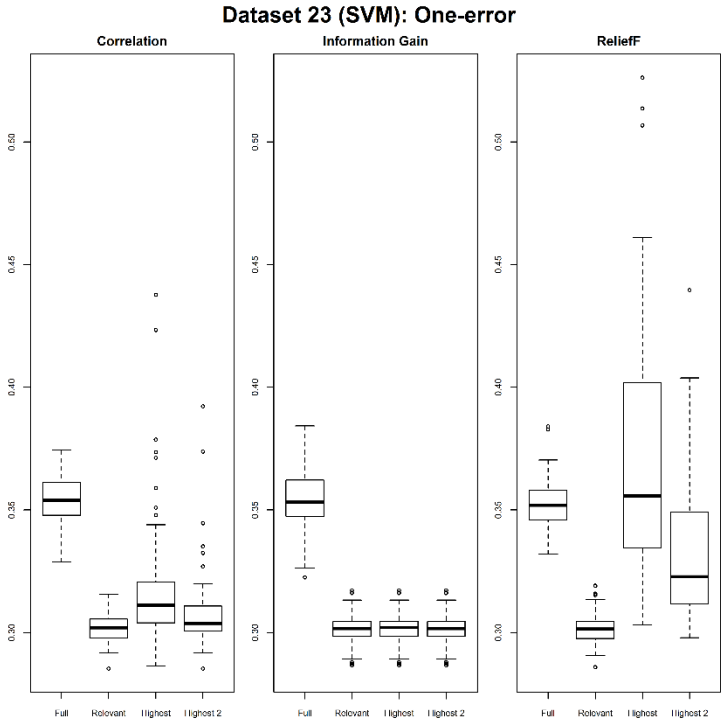


Figure B.90 Comparison of One-error using the SVM classifier: Dataset 23.

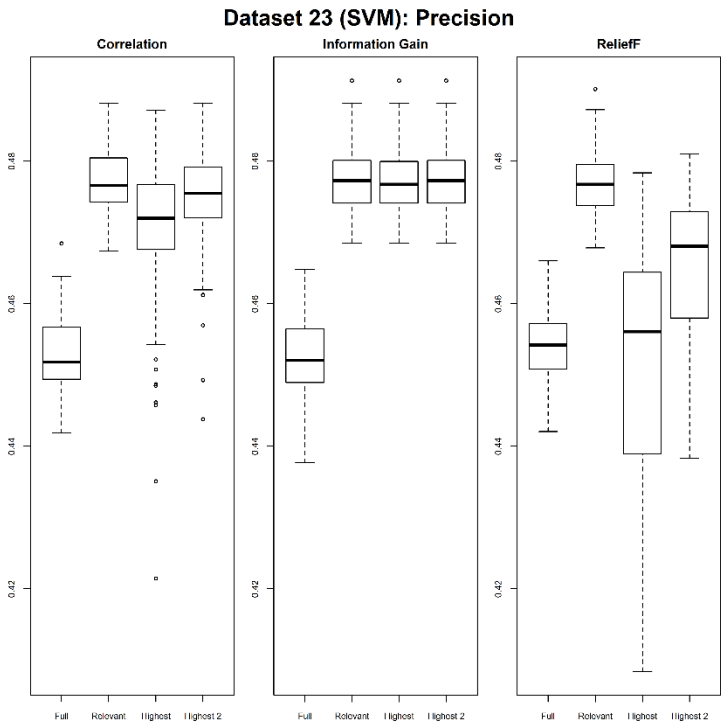


Figure B.91 Comparison of Precision using the SVM classifier: Dataset 23.

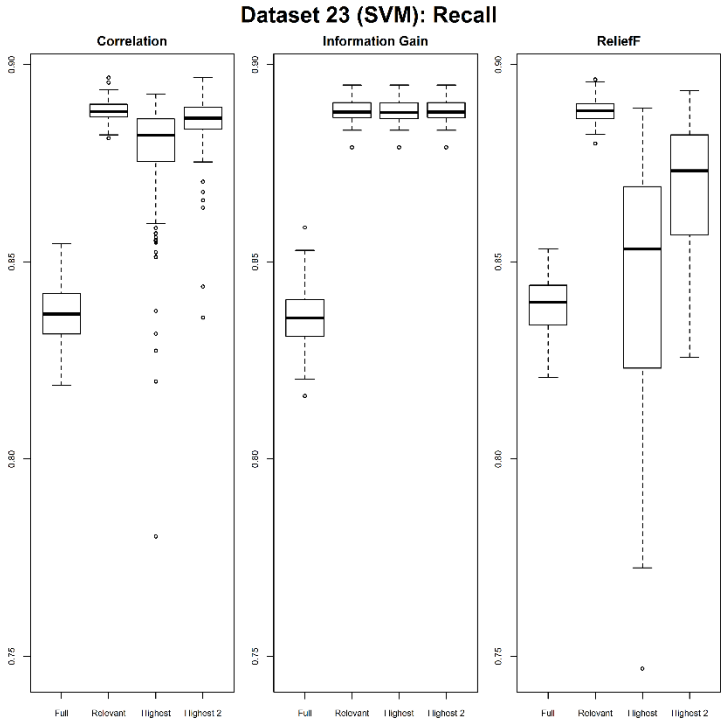


Figure B.92 Comparison of Recall using the SVM classifier: Dataset 23.

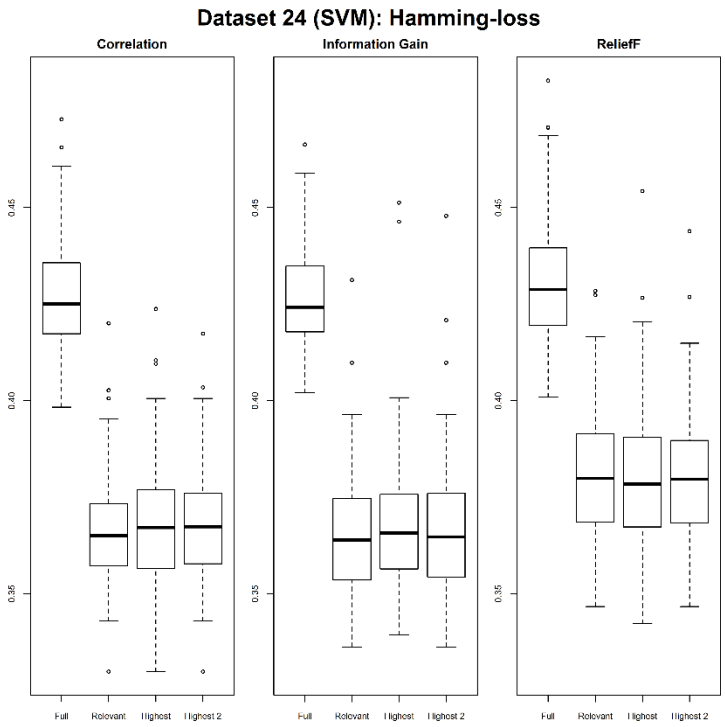


Figure B.93 Comparison of Hamming-loss using the SVM classifier: Dataset 24.

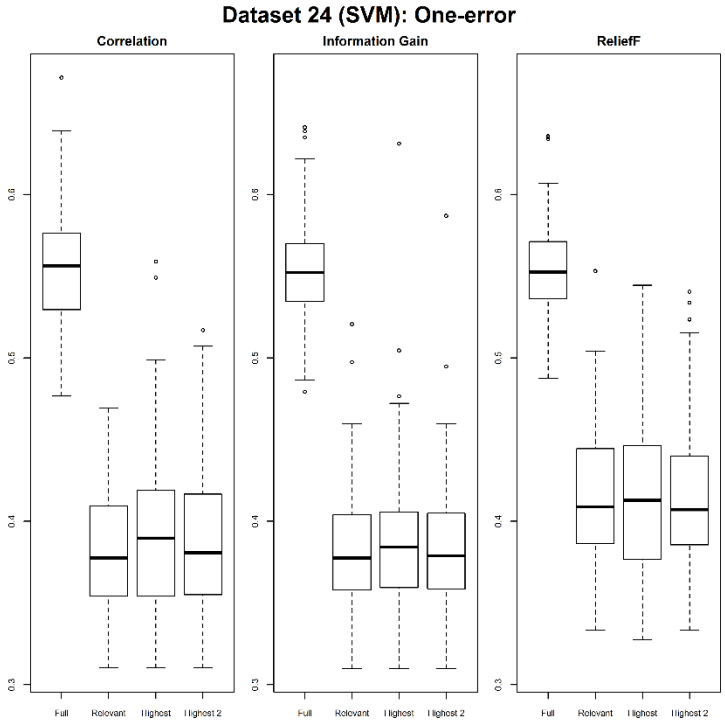


Figure B.94 Comparison of One-error using the SVM classifier: Dataset 24.

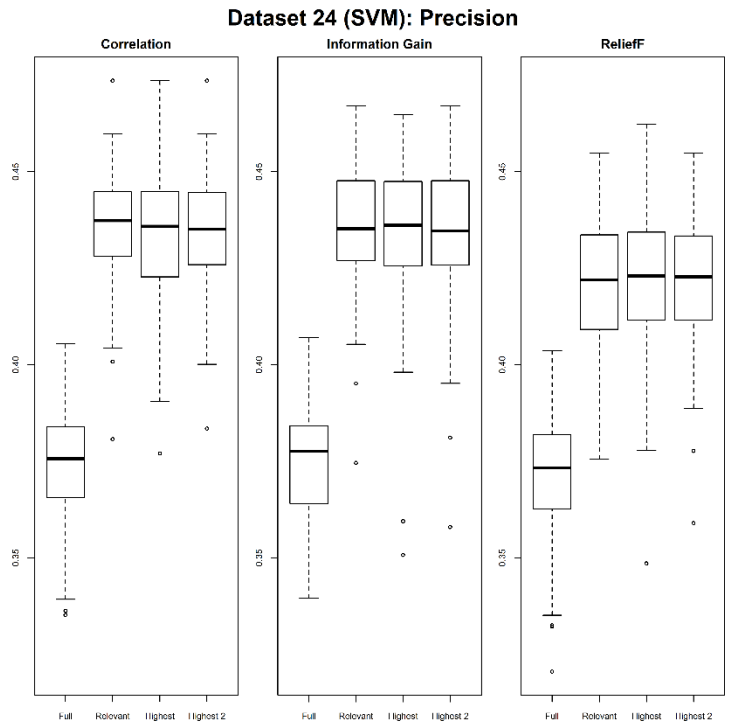


Figure B.95 Comparison of Precision using the SVM classifier: Dataset 24.

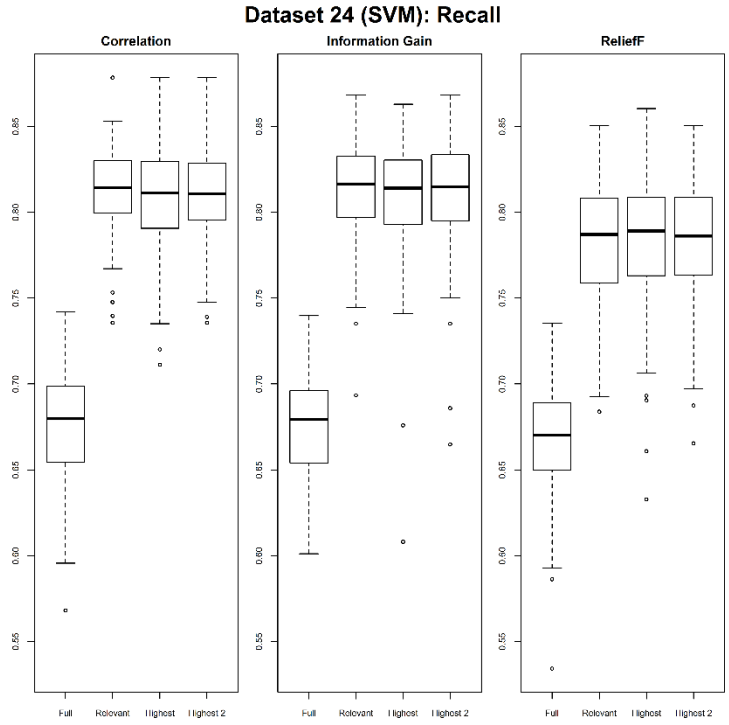


Figure B.96 Comparison of Recall using the SVM classifier: Dataset 24.

APPENDIX C

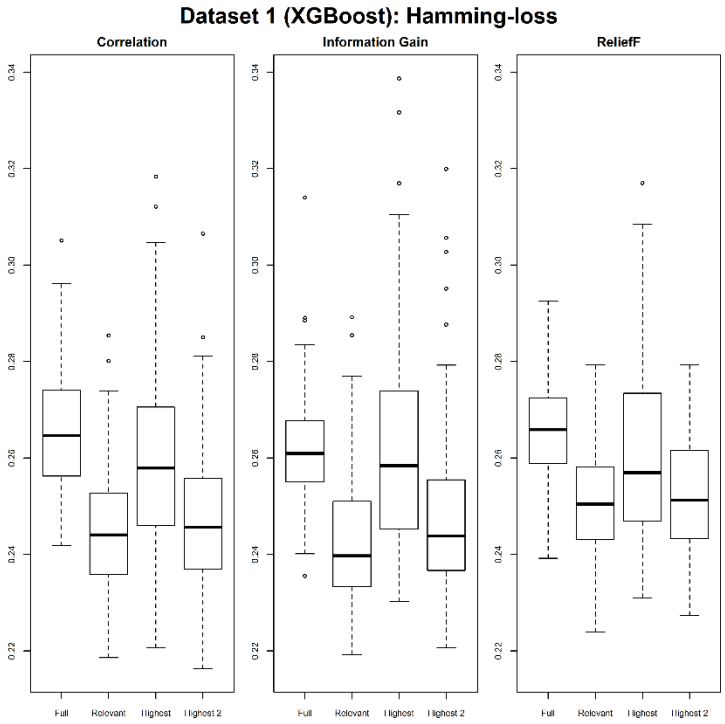


Figure C.1 Comparison of Hamming-loss using the XGBoost classifier: Dataset 1.

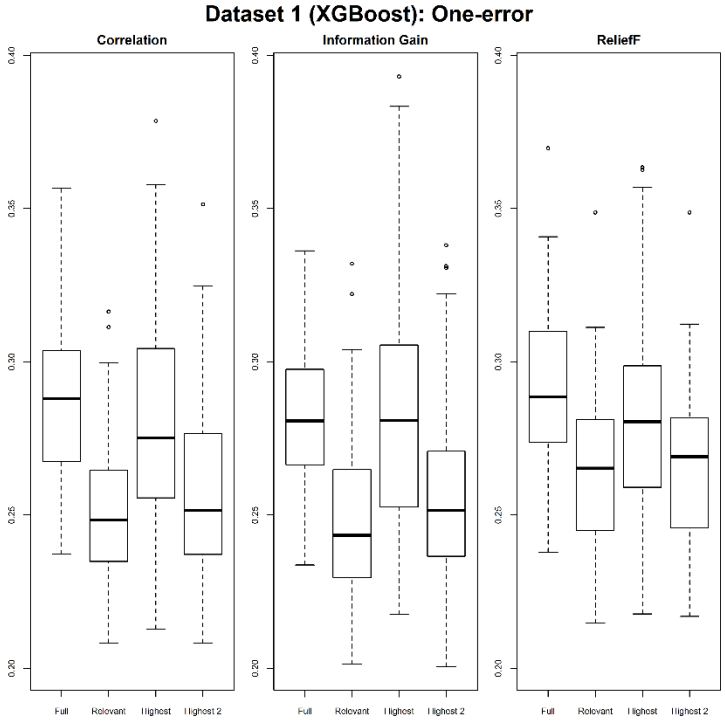


Figure C.2 Comparison of One-error using the XGBoost classifier: Dataset 1.

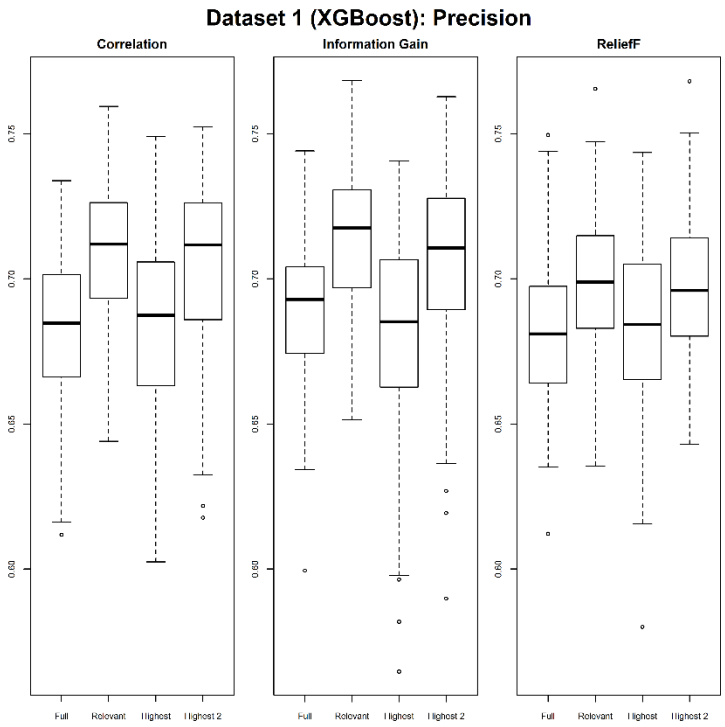


Figure C.3 Comparison of Precision using the XGBoost classifier: Dataset 1.

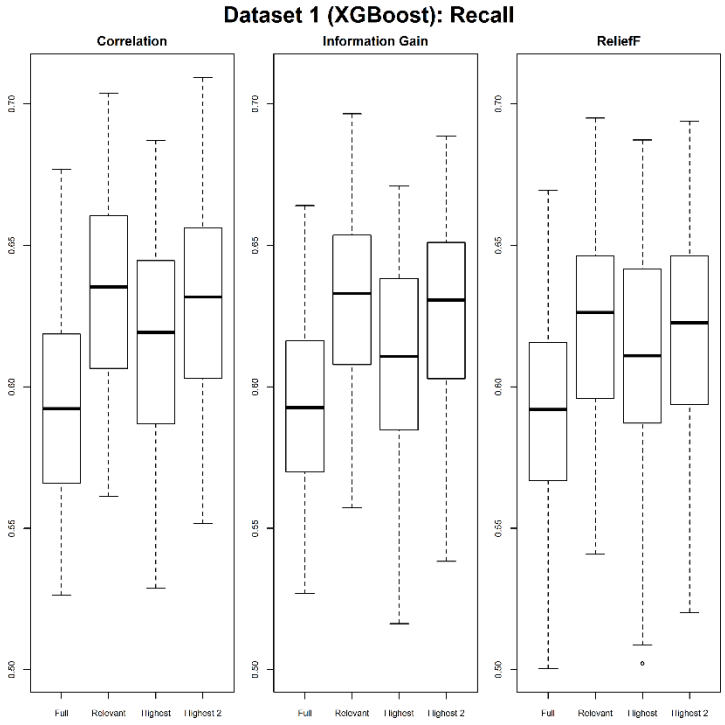


Figure C.4 Comparison of Recall using the XGBoost classifier: Dataset 1.

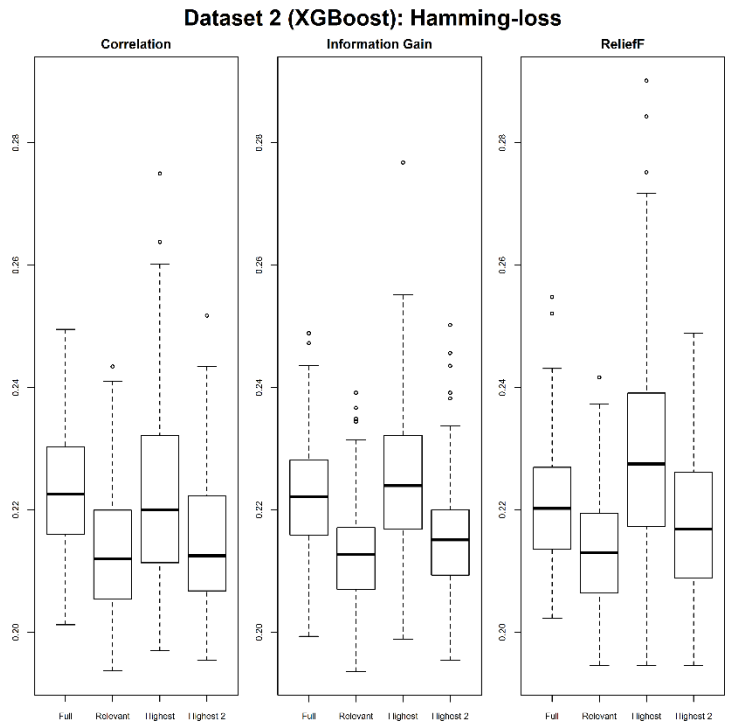


Figure C.5 Comparison of Hamming-loss using the XGBoost classifier: Dataset 2.

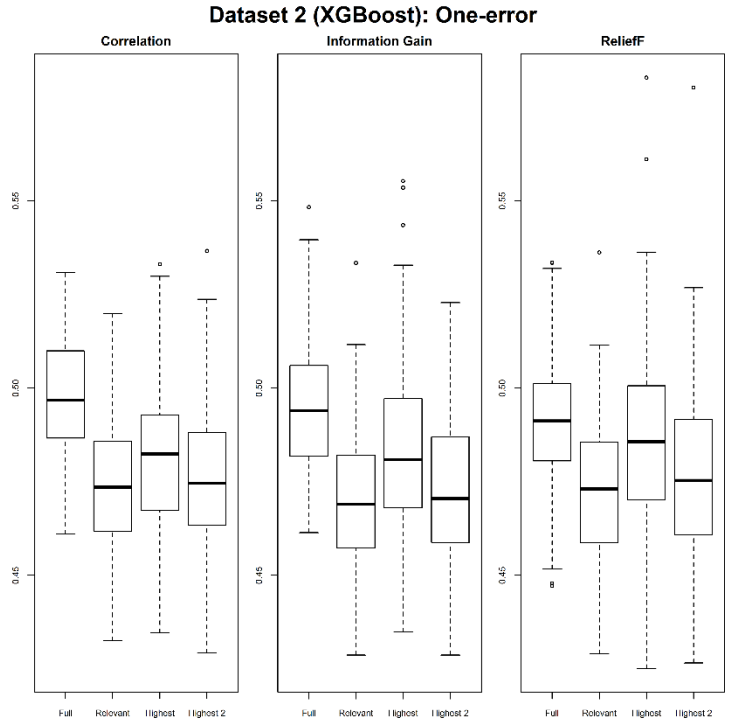


Figure C.6 Comparison of One-error using the XGBoost classifier: Dataset 2.

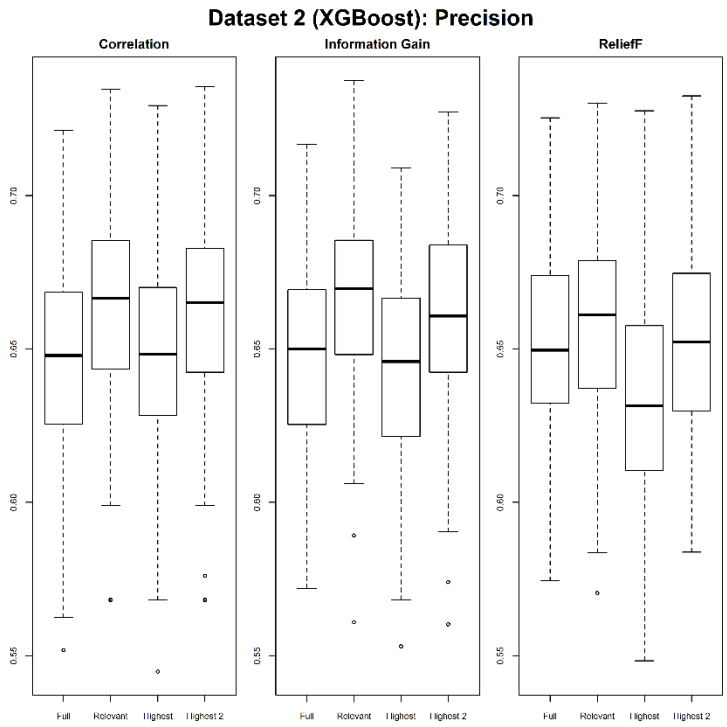


Figure C.7 Comparison of Precision using the XGBoost classifier: Dataset 2.

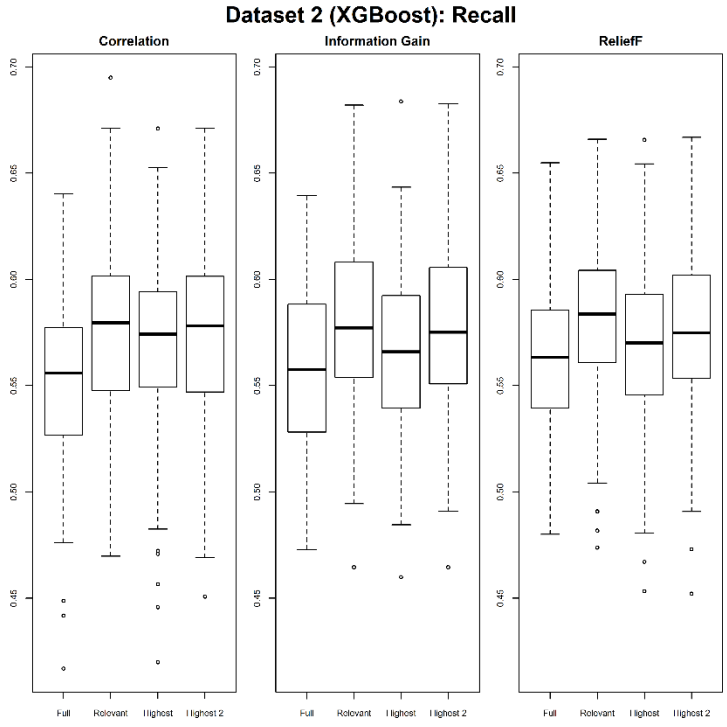


Figure C.8 Comparison of Recall using the XGBoost classifier: Dataset 2.

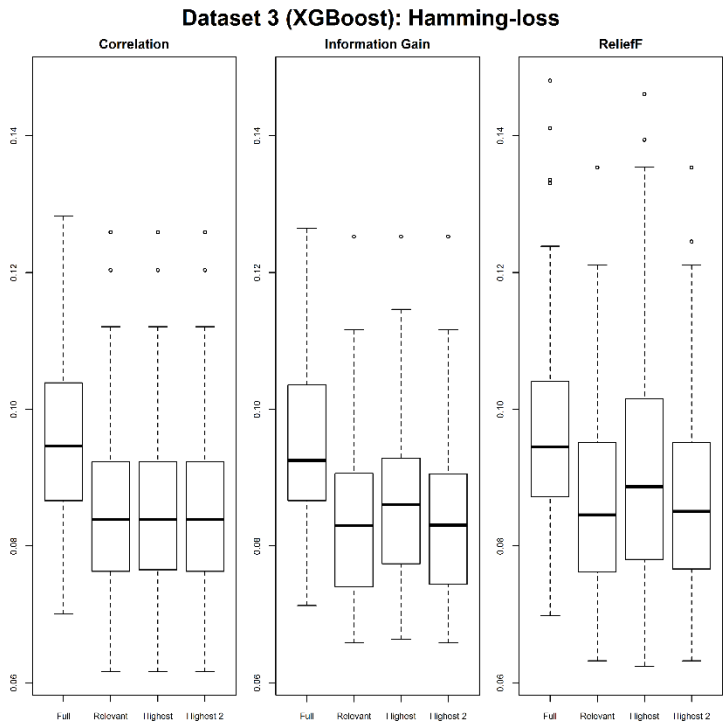


Figure C.9 Comparison of Hamming-loss using the XGBoost classifier: Dataset 3.

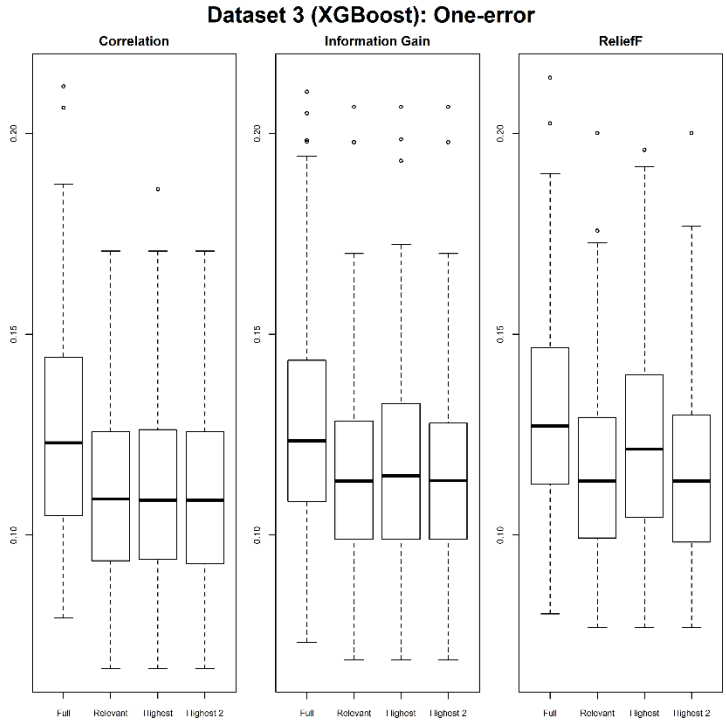


Figure C.10 Comparison of One-error using the XGBoost classifier: Dataset 3.

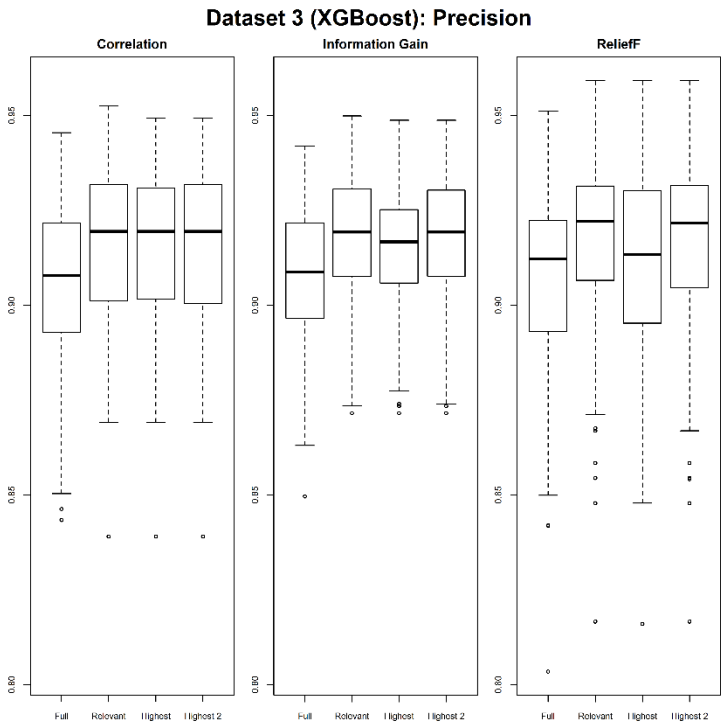


Figure C.11 Comparison of Precision using the XGBoost classifier: Dataset 3.

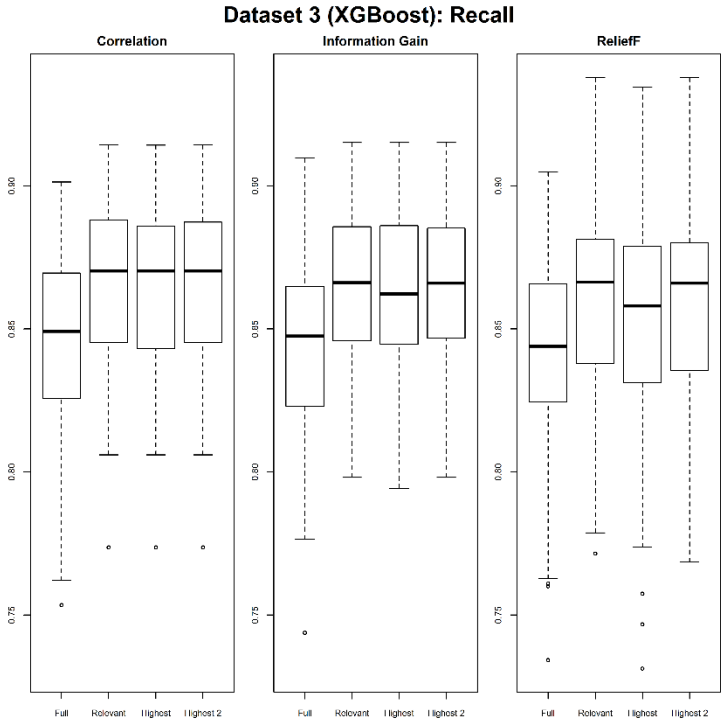


Figure C.12 Comparison of Recall using the XGBoost classifier: Dataset 3.

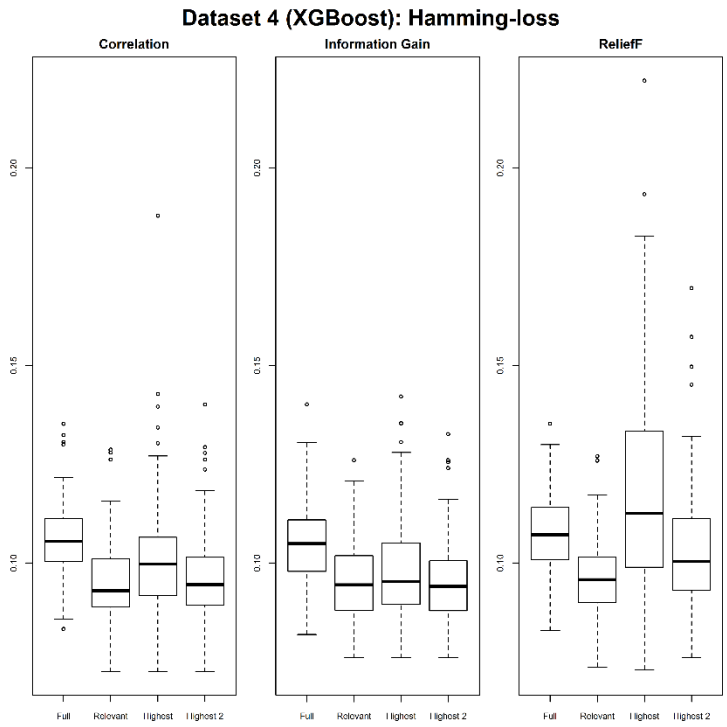


Figure C.13 Comparison of Hamming-loss using the XGBoost classifier: Dataset 4.

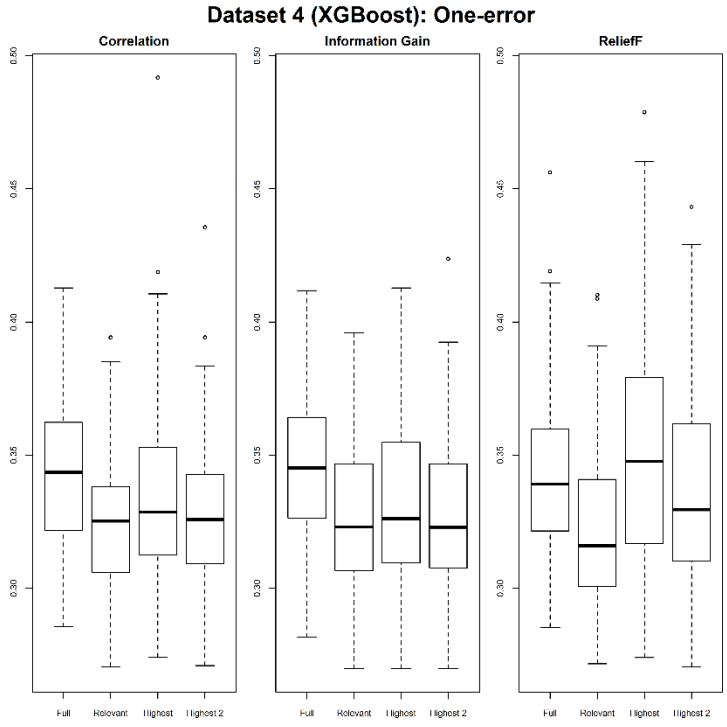


Figure C.14 Comparison of One-error using the XGBoost classifier: Dataset 4.

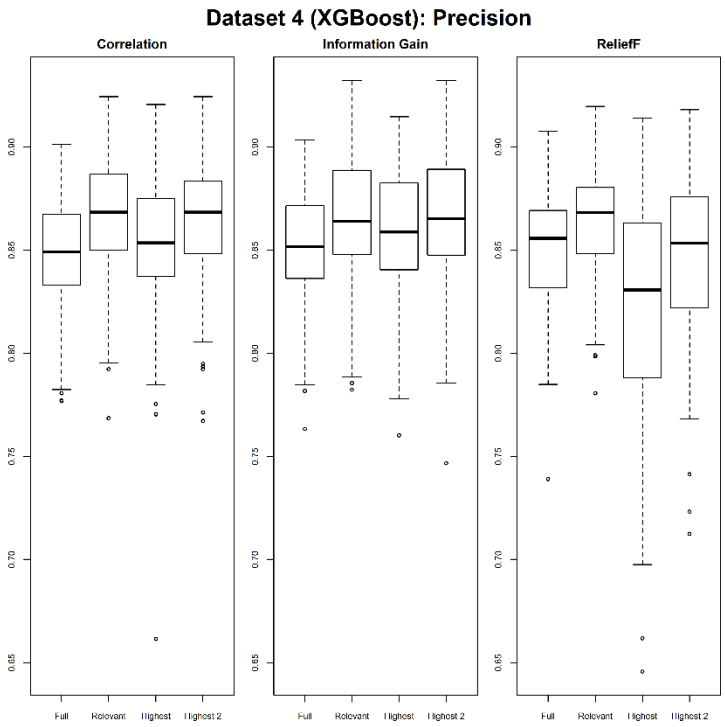


Figure C.15 Comparison of Precision using the XGBoost classifier: Dataset 4.

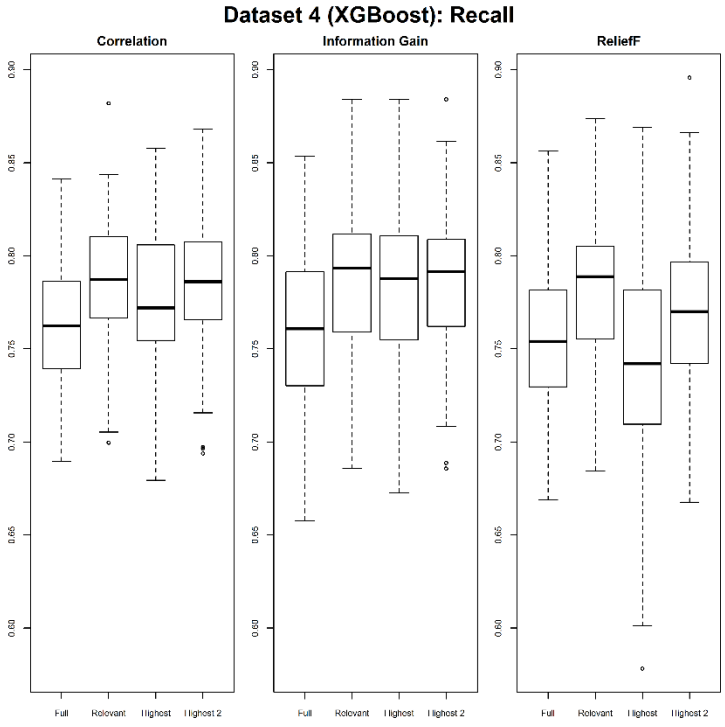


Figure C.16 Comparison of Recall using the XGBoost classifier: Dataset 4.

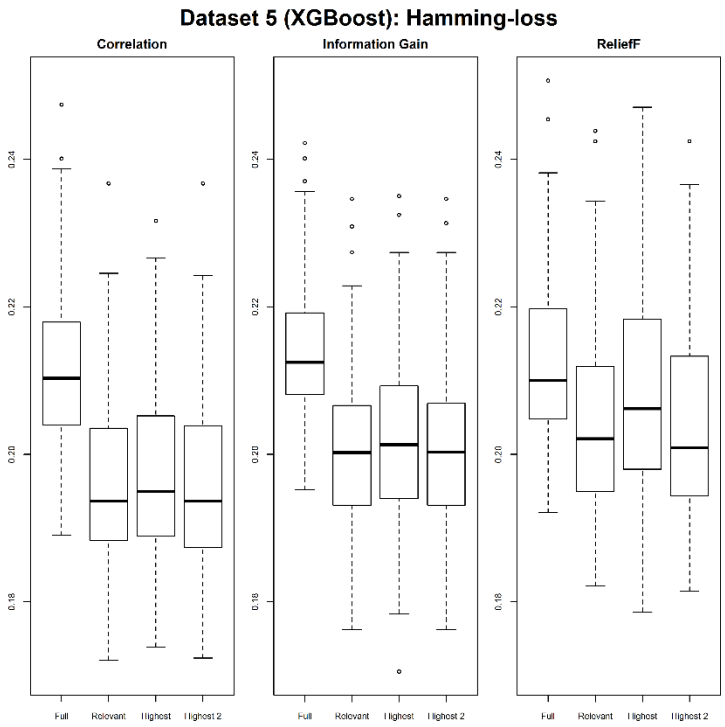


Figure C.17 Comparison of Hamming-loss using the XGBoost classifier: Dataset 5.

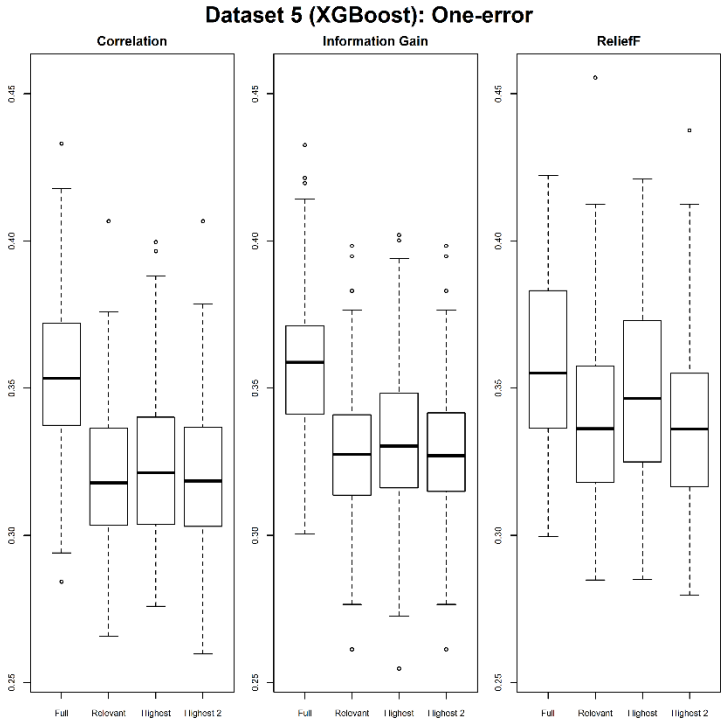


Figure C.18 Comparison of One-error using the XGBoost classifier: Dataset 5.

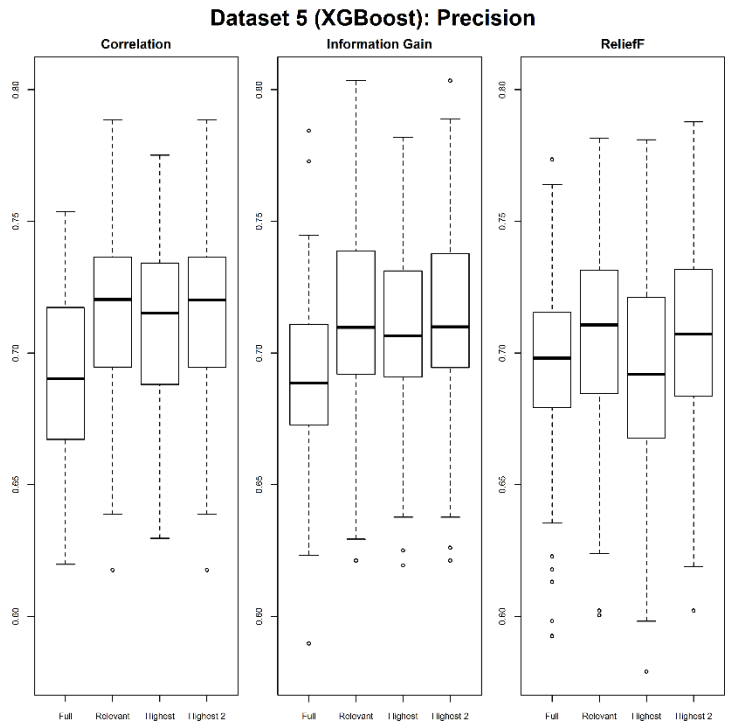


Figure C.19 Comparison of Precision using the XGBoost classifier: Dataset 5.

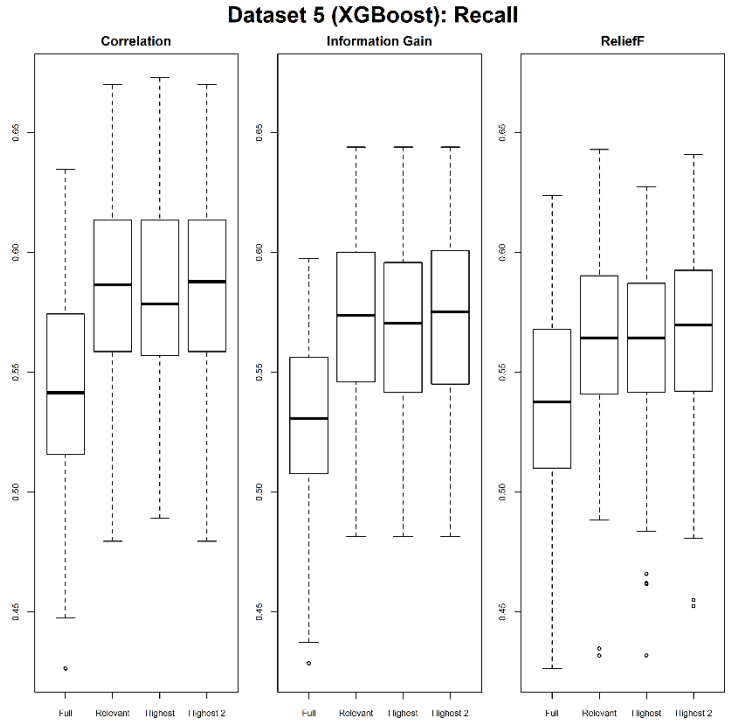


Figure C.20 Comparison of Recall using the XGBoost classifier: Dataset 5.

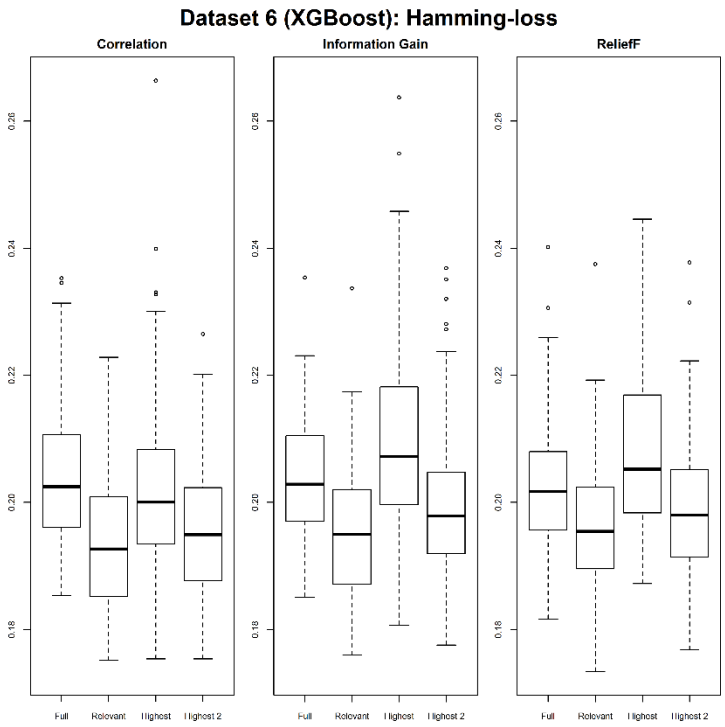


Figure C.21 Comparison of Hamming-loss using the XGBoost classifier: Dataset 6.

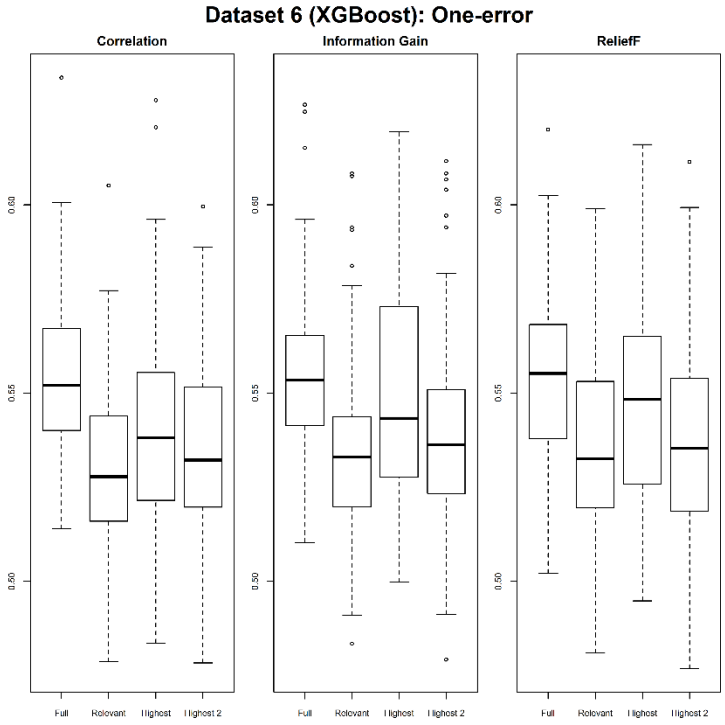


Figure C.22 Comparison of One-error using the XGBoost classifier: Dataset 6.

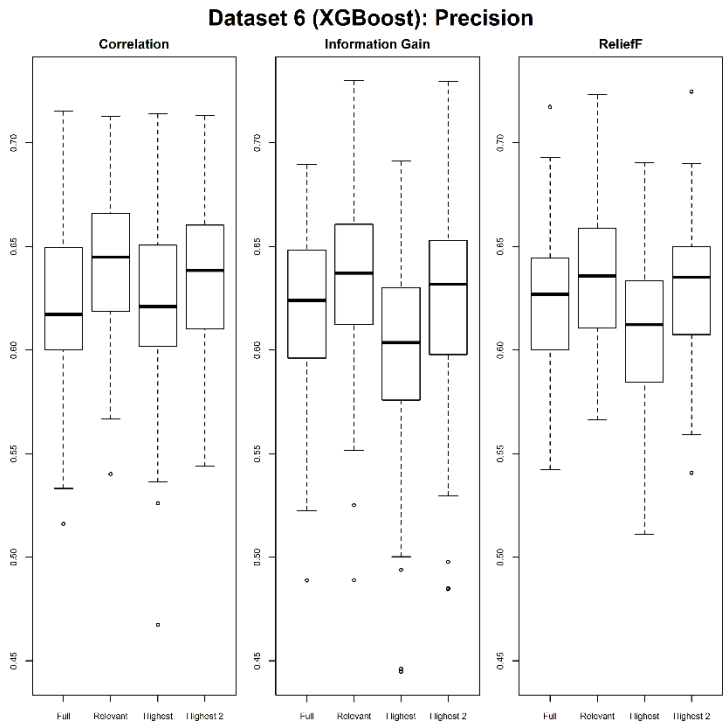


Figure C.23 Comparison of Precision using the XGBoost classifier: Dataset 6.

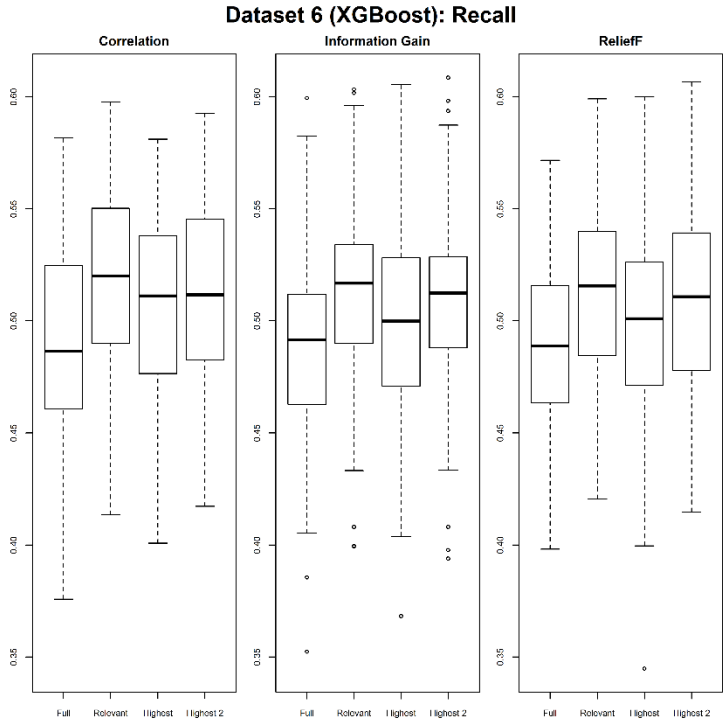


Figure C.24 Comparison of Recall using the XGBoost classifier: Dataset 6.

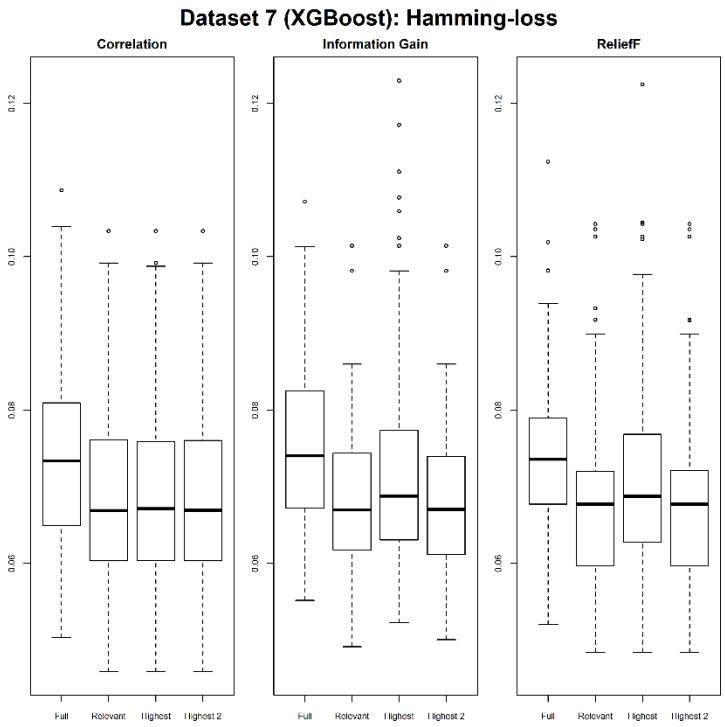


Figure C.25 Comparison of Hamming-loss using the XGBoost classifier: Dataset 7.

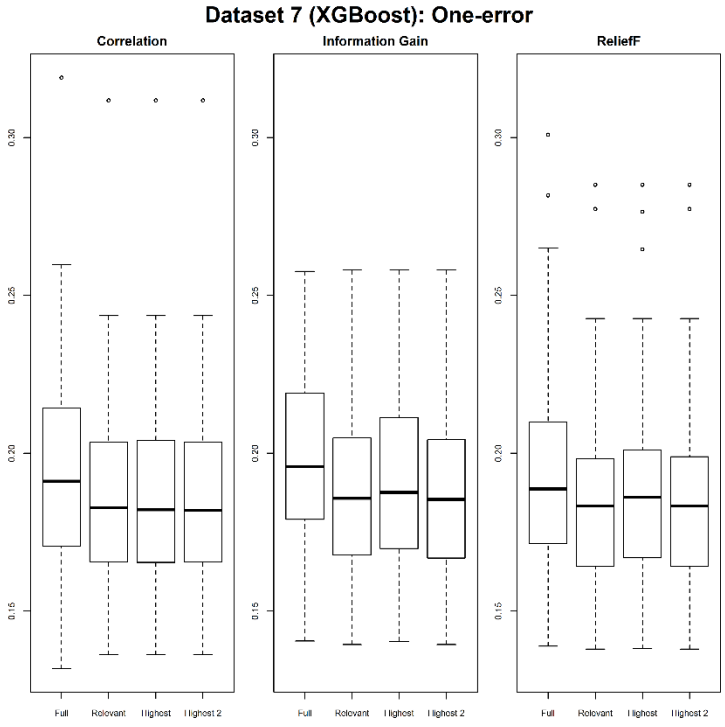


Figure C.26 Comparison of One-error using the XGBoost classifier: Dataset 7.

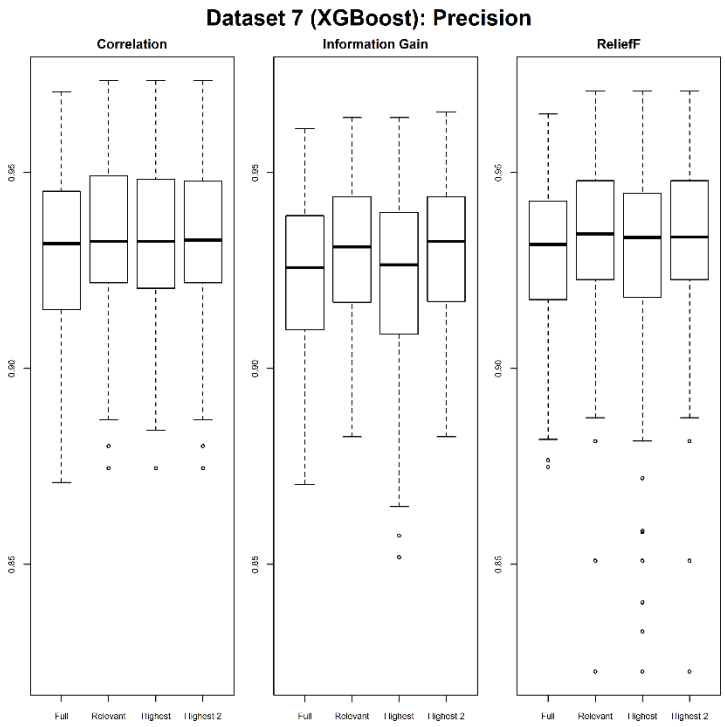


Figure C.27 Comparison of Precision using the XGBoost classifier: Dataset 7.

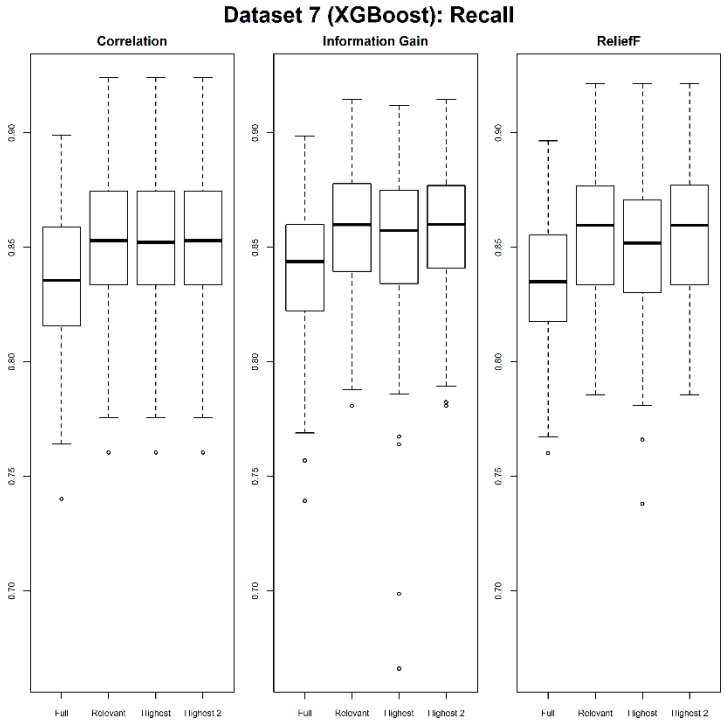


Figure C.28 Comparison of Recall using the XGBoost classifier: Dataset 7.

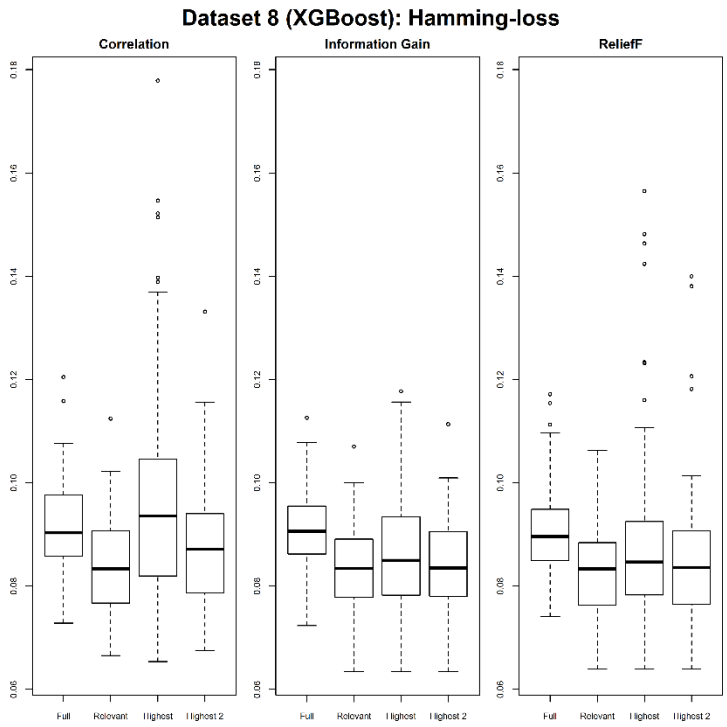


Figure C.29 Comparison of Hamming-loss using the XGBoost classifier: Dataset 8.

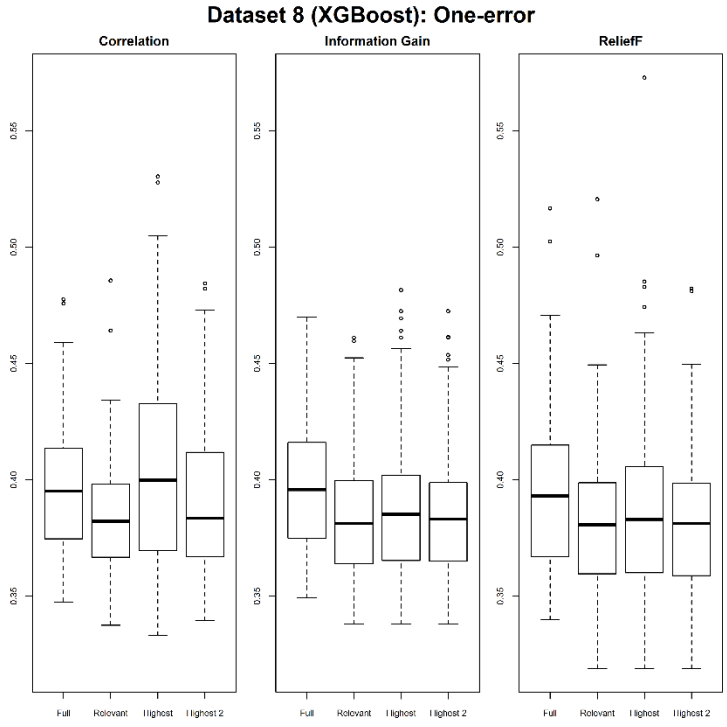


Figure C.30 Comparison of One-error using the XGBoost classifier: Dataset 8.

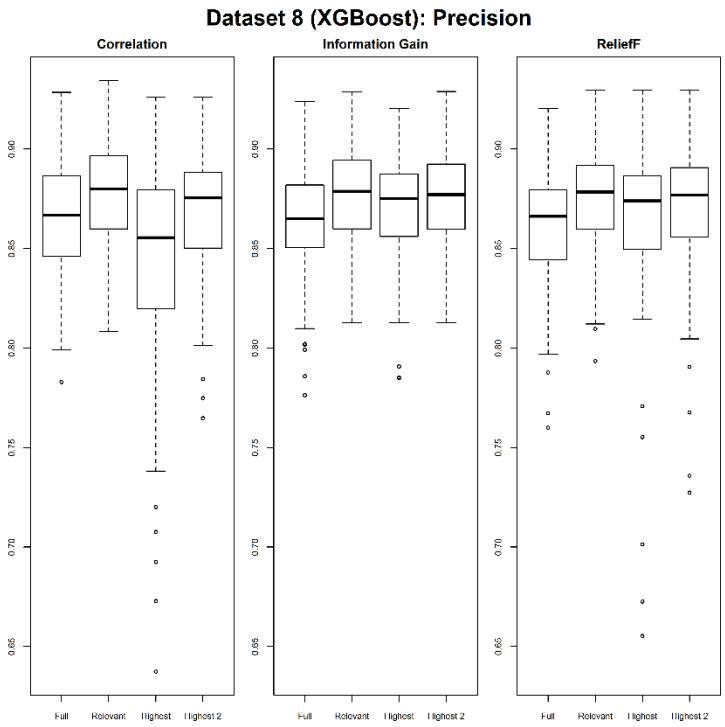


Figure C.31 Comparison of Precision using the XGBoost classifier: Dataset 8.

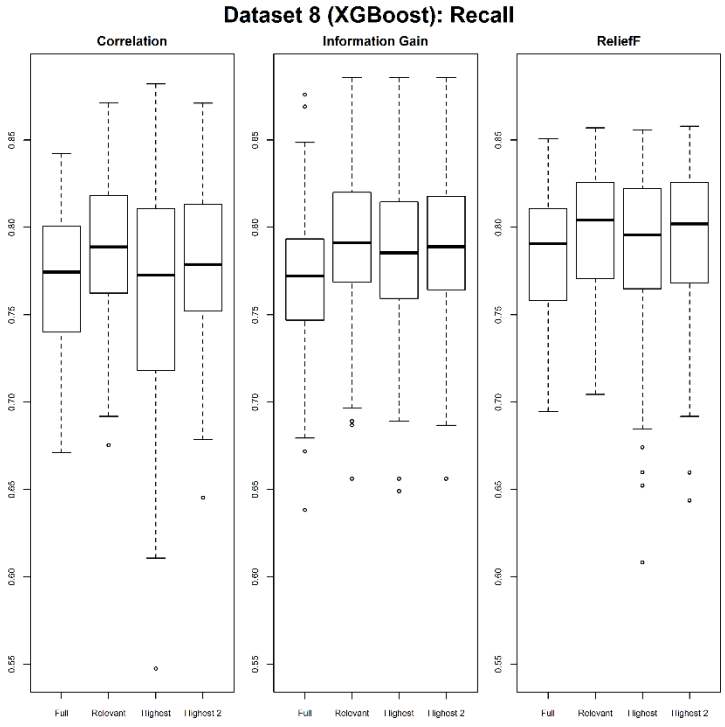


Figure C.32 Comparison of Recall using the XGBoost classifier: Dataset 8.

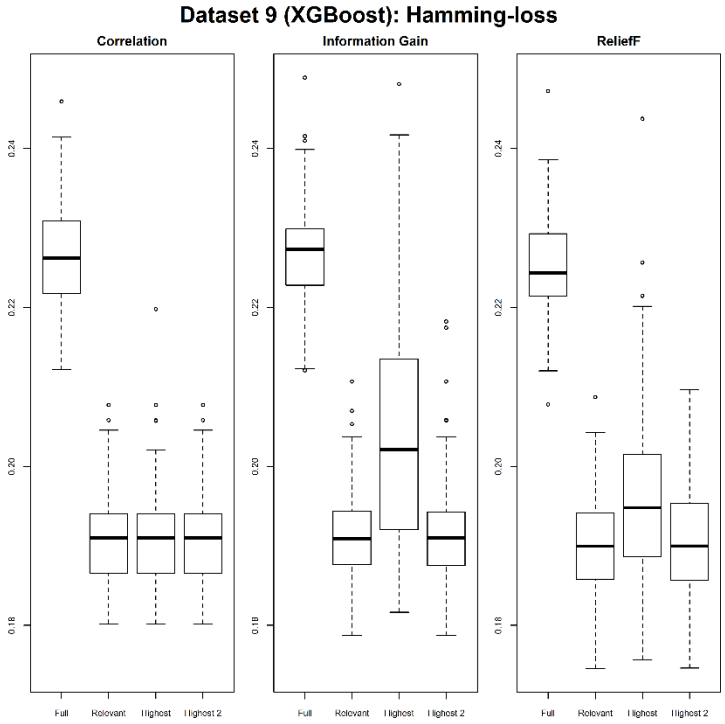


Figure C.33 Comparison of Hamming-loss using the XGBoost classifier: Dataset 9.

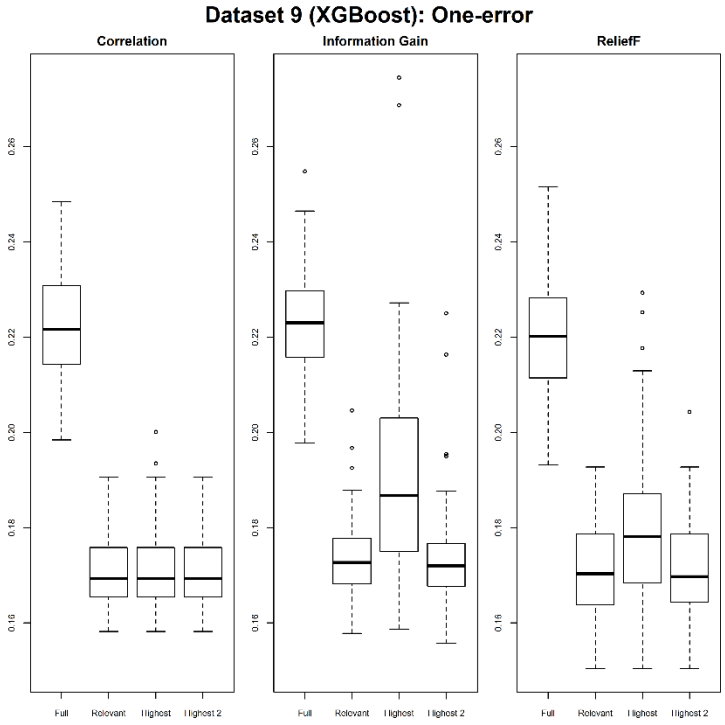


Figure C.34 Comparison of One-error using the XGBoost classifier: Dataset 9.

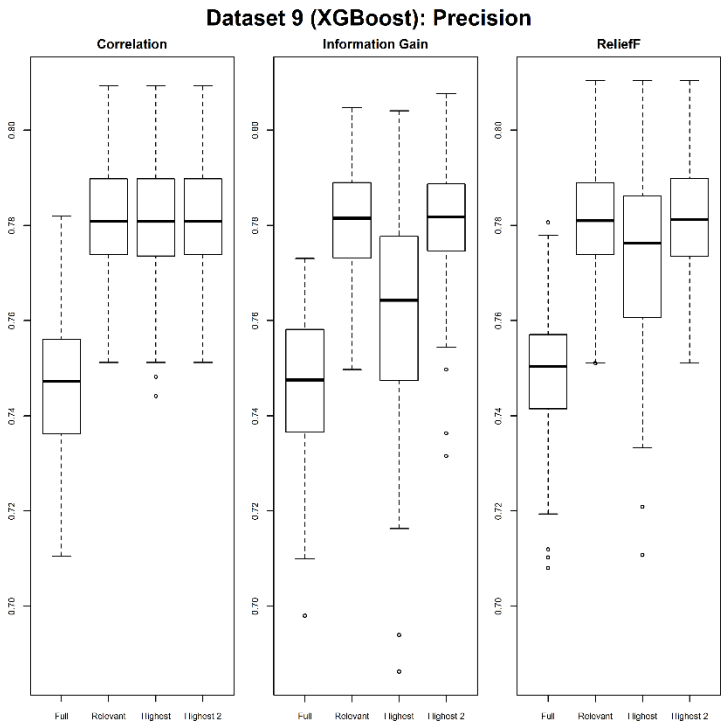


Figure C.35 Comparison of Precision using the XGBoost classifier: Dataset 9.

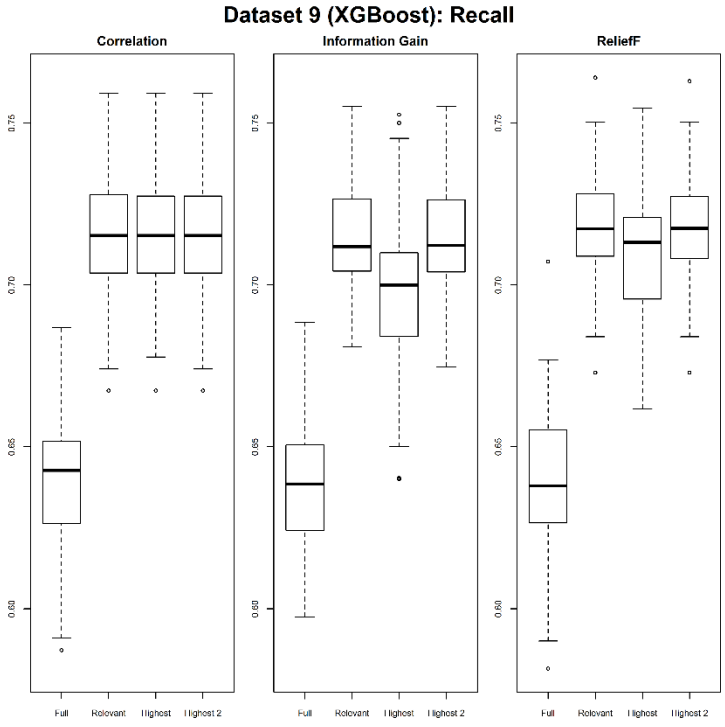


Figure C.36 Comparison of Recall using the XGBoost classifier: Dataset 9.

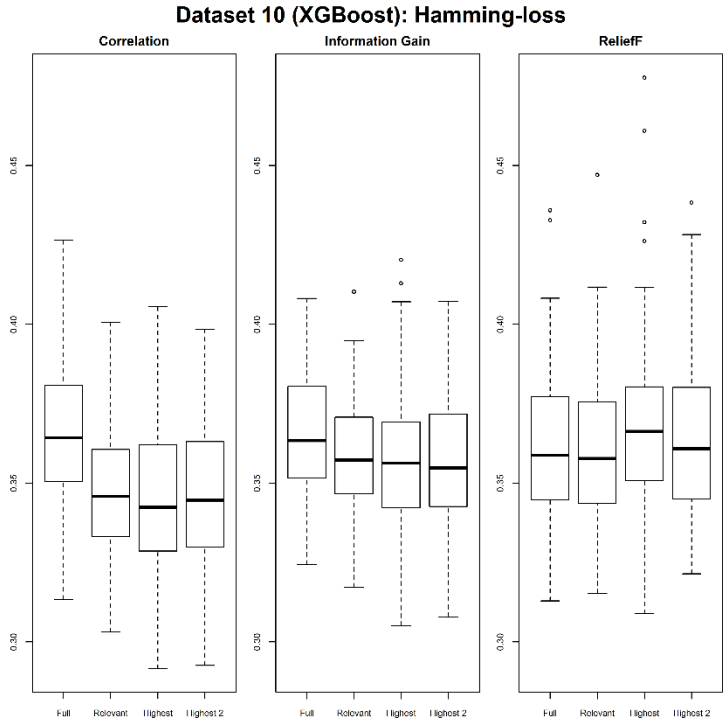


Figure C.37 Comparison of Hamming-loss using the XGBoost classifier: Dataset 10.

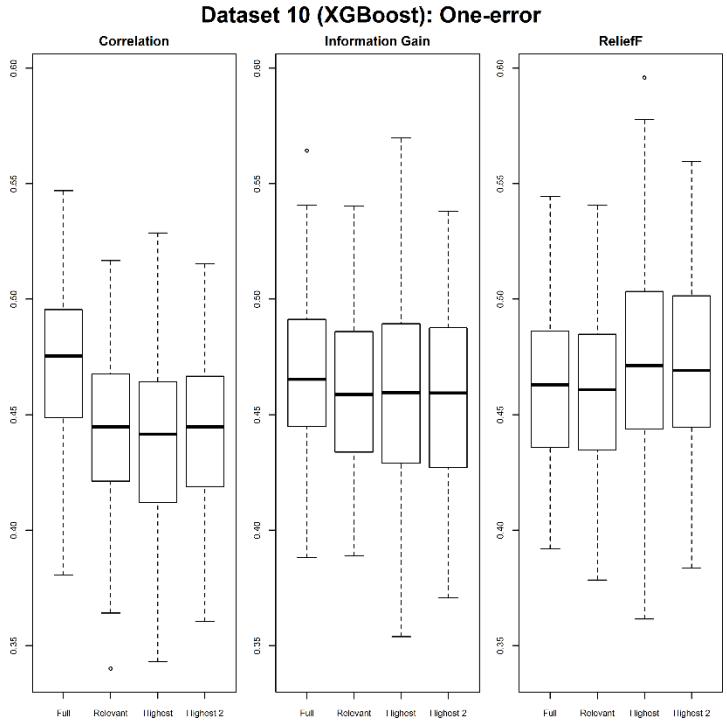


Figure C.38 Comparison of One-error using the XGBoost classifier: Dataset 10.

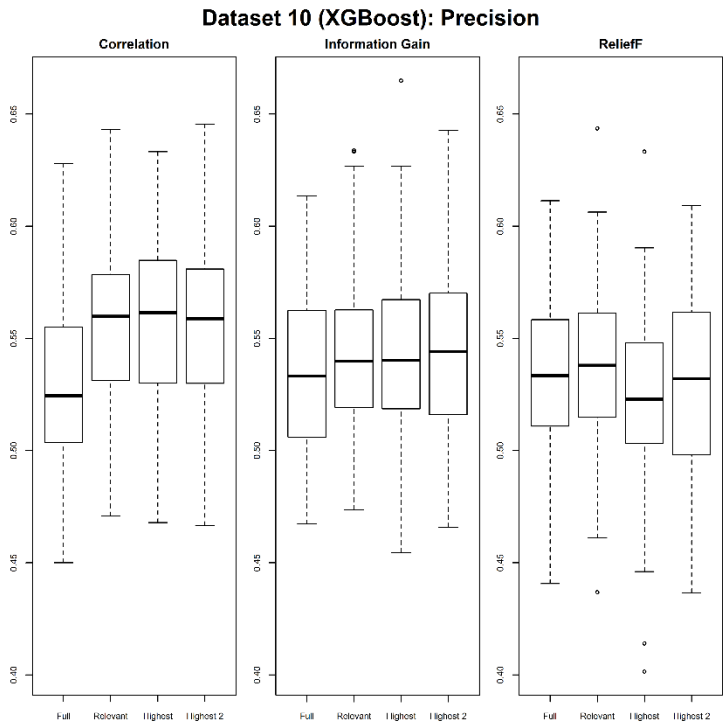


Figure C.39 Comparison of Precision using the XGBoost classifier: Dataset 10.

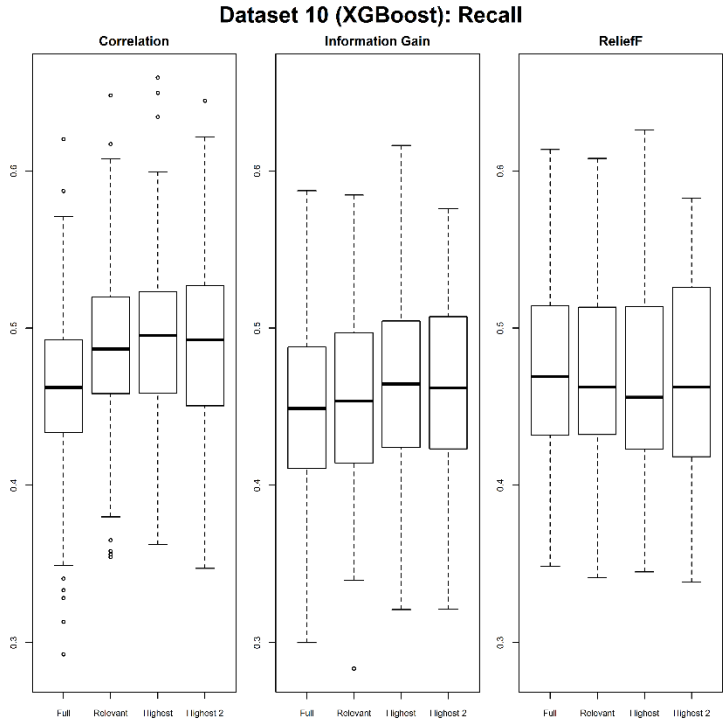


Figure C.40 Comparison of Recall using the XGBoost classifier: Dataset 10.

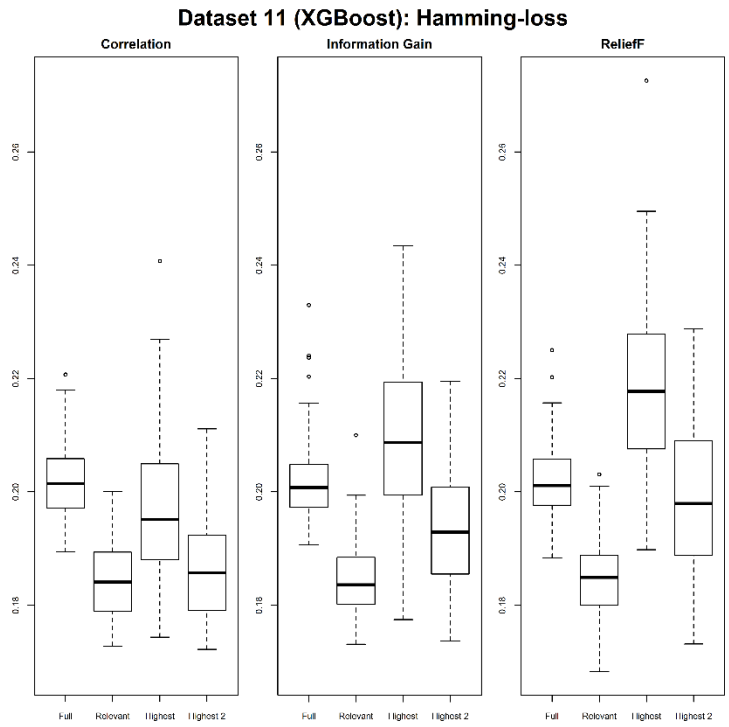


Figure C.41 Comparison of Hamming-loss using the XGBoost classifier: Dataset 11.

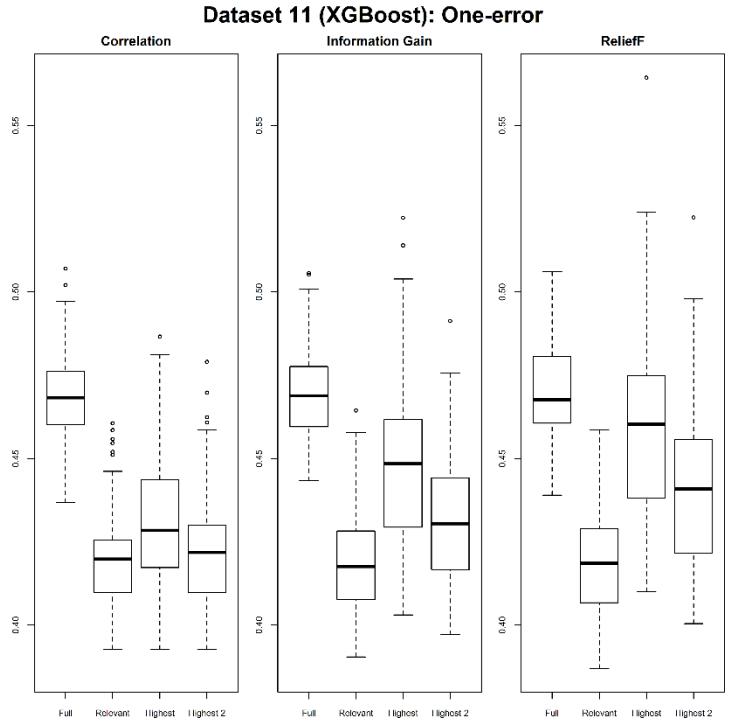


Figure C.42 Comparison of One-error using the XGBoost classifier: Dataset 11.

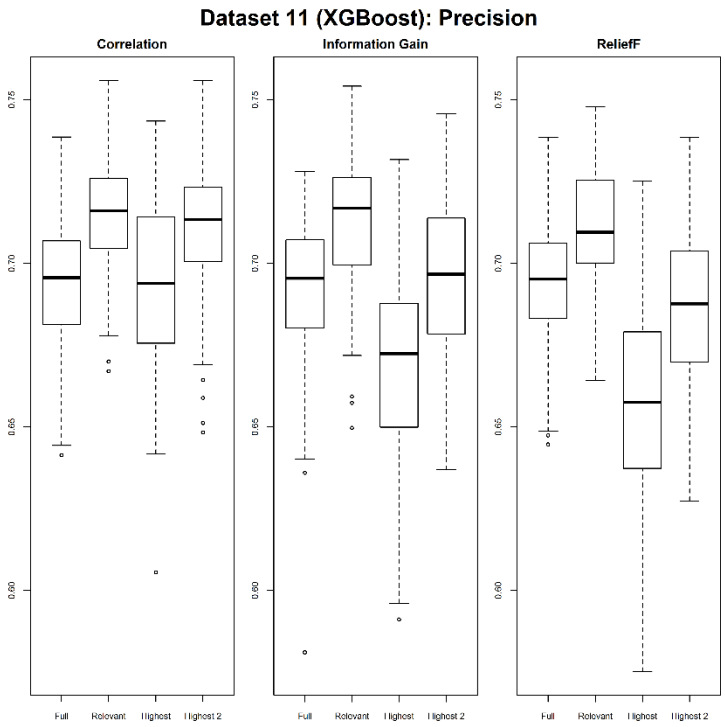


Figure C.43 Comparison of Precision using the XGBoost classifier: Dataset 11.

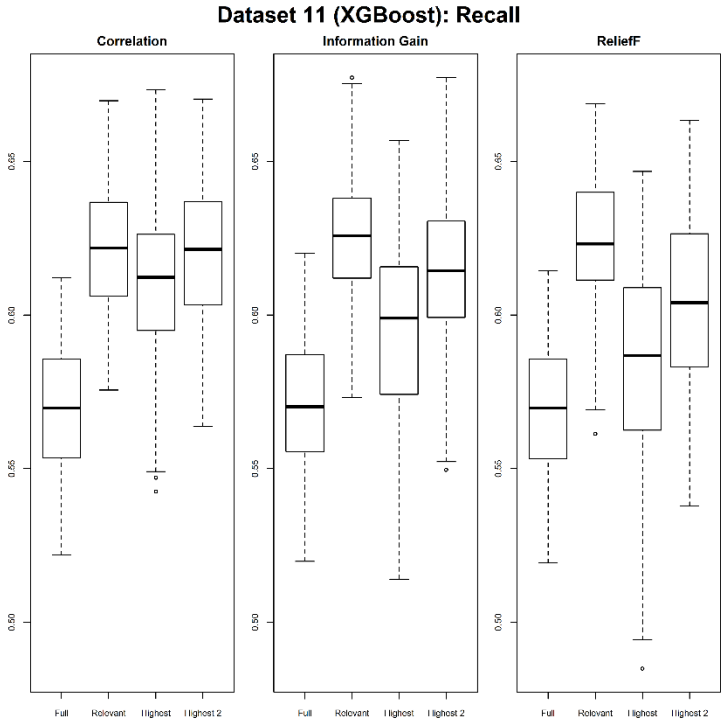


Figure C.44 Comparison of Recall using the XGBoost classifier: Dataset 11.

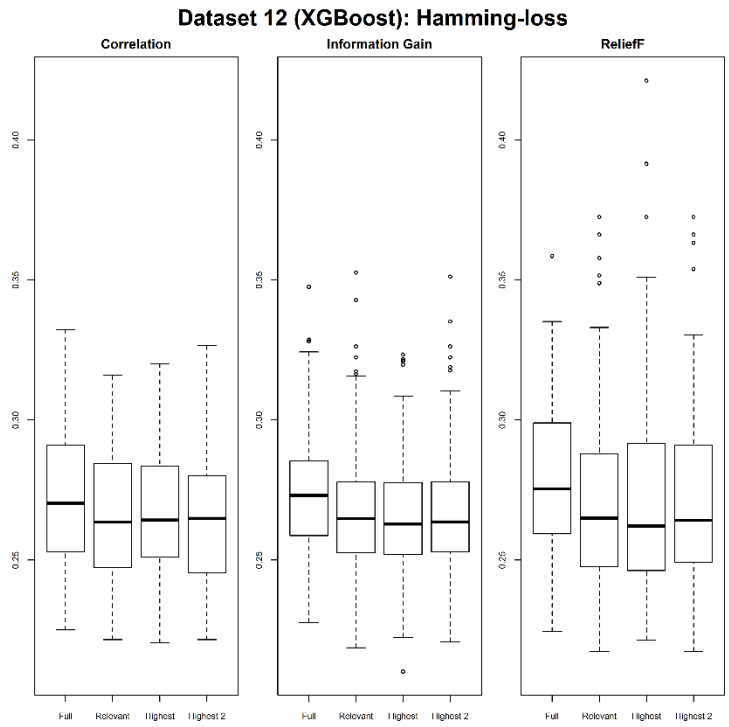


Figure C.45 Comparison of Hamming-loss using the XGBoost classifier: Dataset 12.

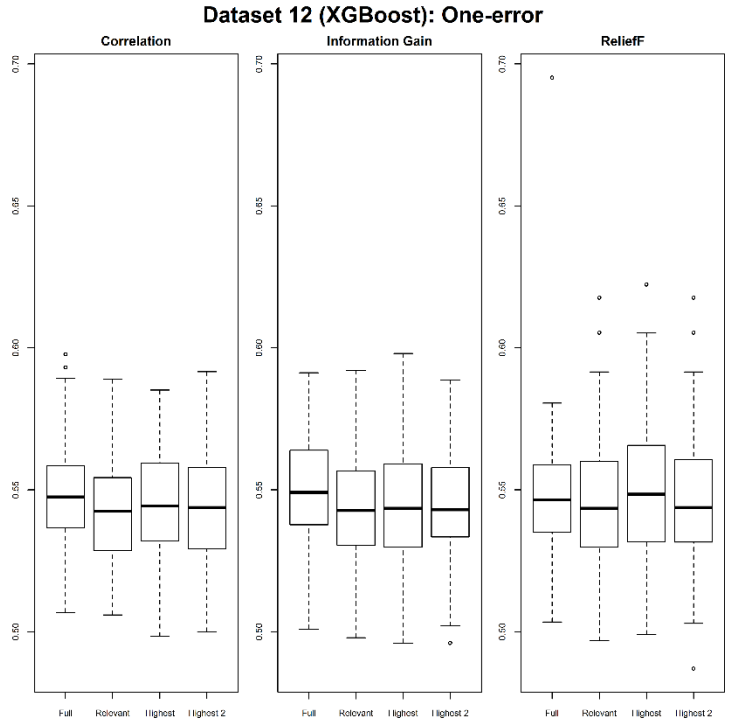


Figure C.46 Comparison of One-error using the XGBoost classifier: Dataset 12.

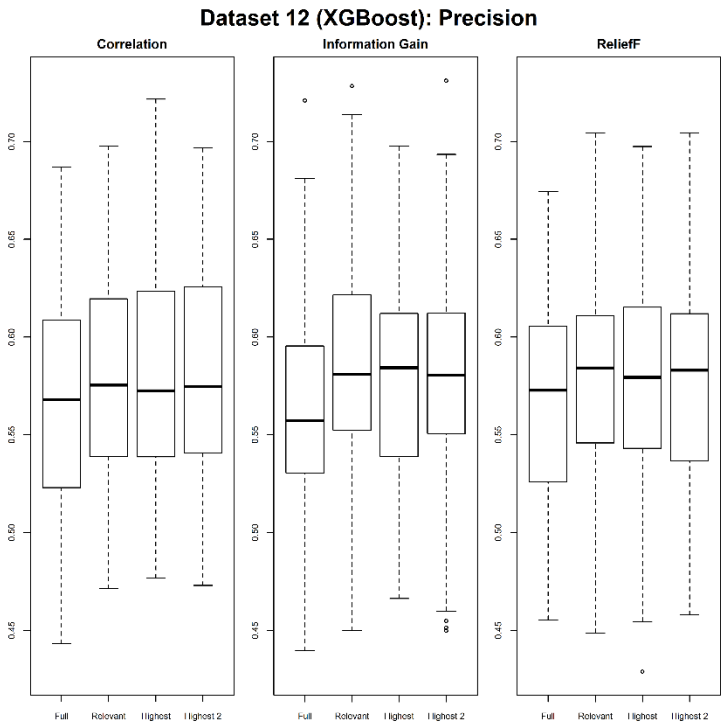


Figure C.47 Comparison of Precision using the XGBoost classifier: Dataset 12.

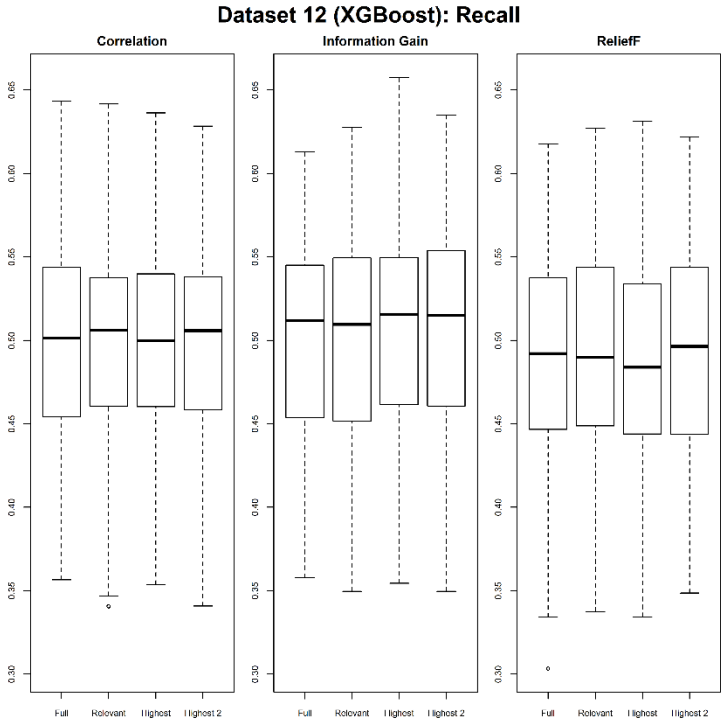


Figure C.48 Comparison of Recall using the XGBoost classifier: Dataset 12.

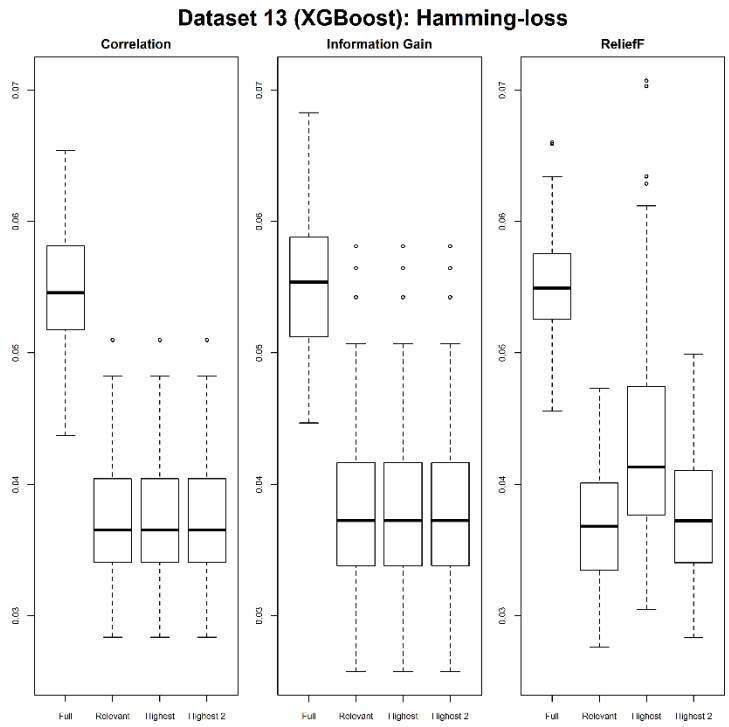


Figure C.49 Comparison of Hamming-loss using the XGBoost classifier: Dataset 13.

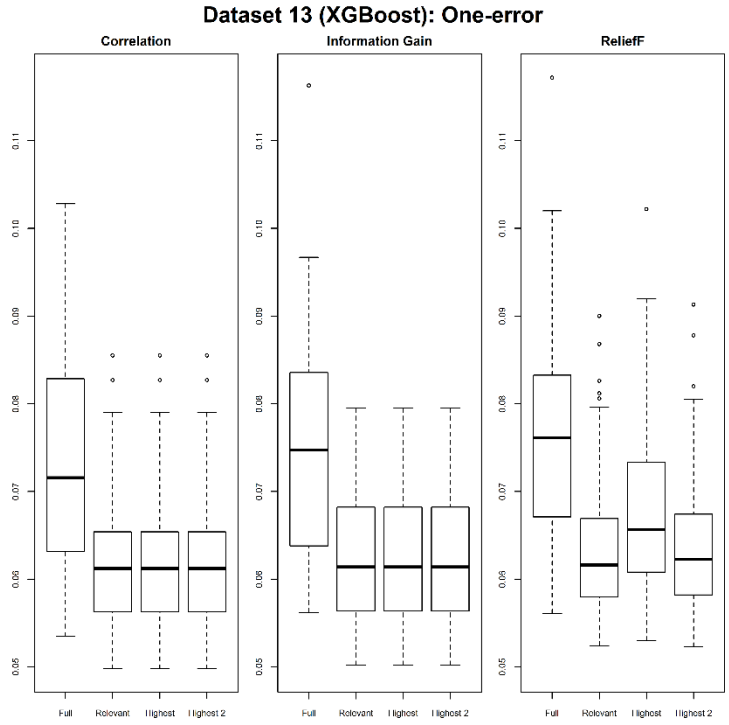


Figure C.50 Comparison of One-error using the XGBoost classifier: Dataset 13.

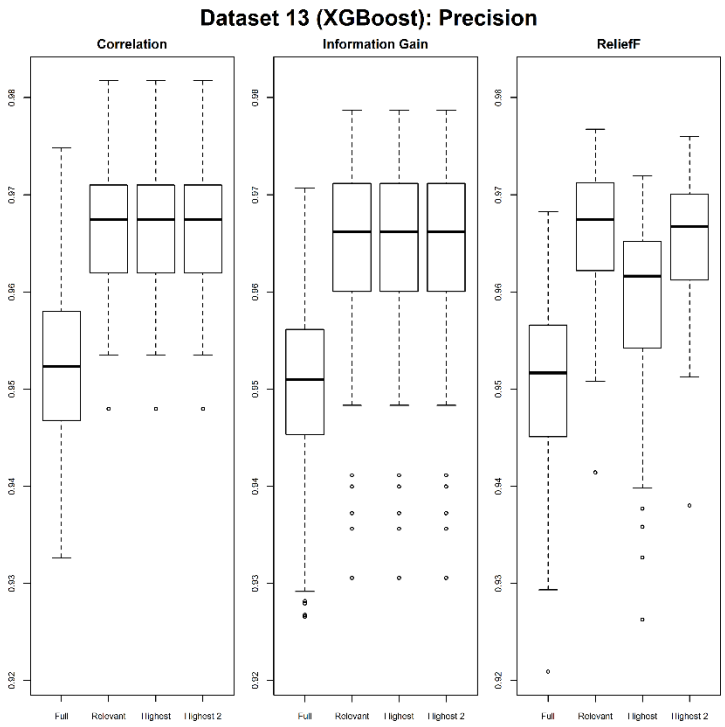


Figure C.51 Comparison of Precision using the XGBoost classifier: Dataset 13.

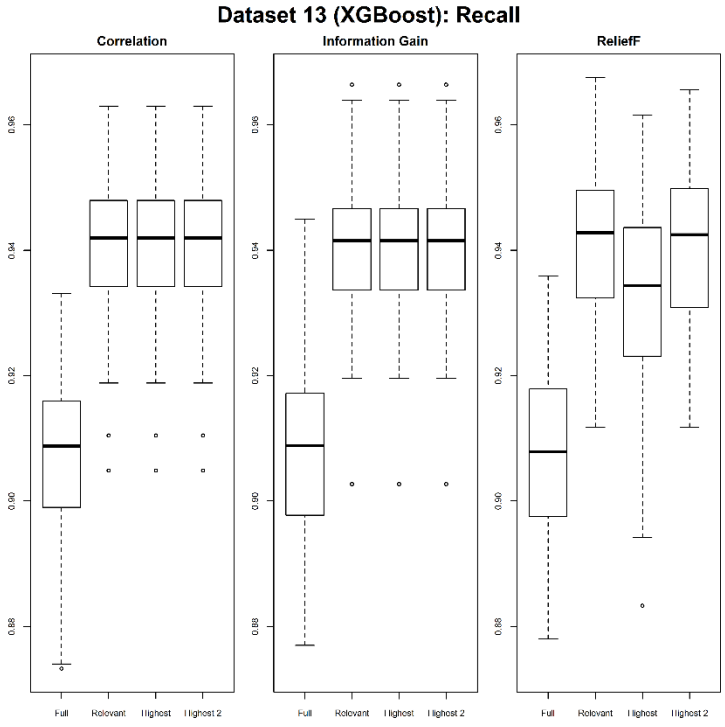


Figure C.52 Comparison of Recall using the XGBoost classifier: Dataset 13.

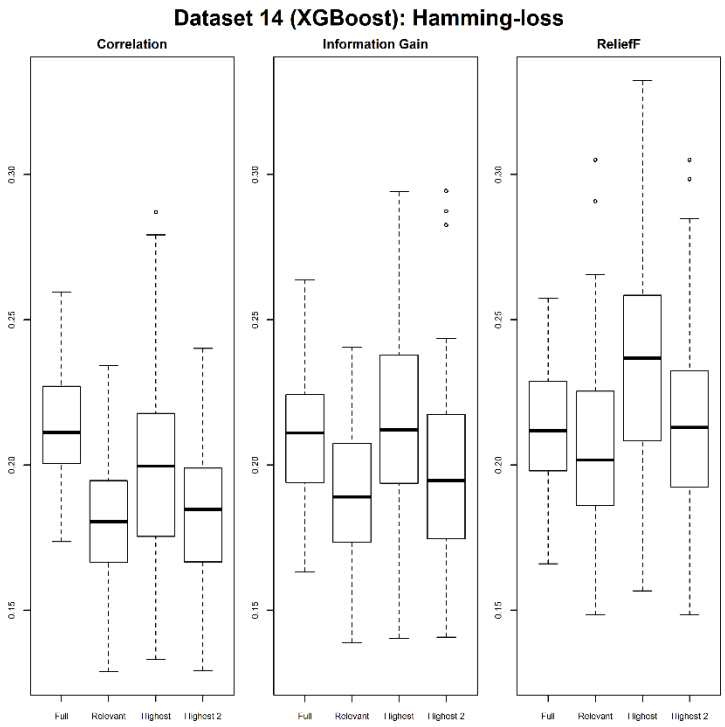


Figure C.53 Comparison of Hamming-loss using the XGBoost classifier: Dataset 14.

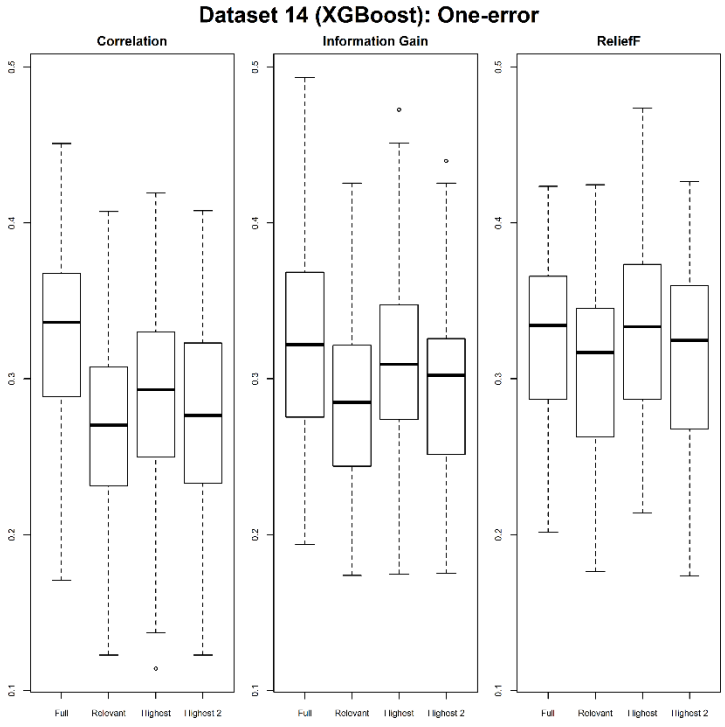


Figure C.54 Comparison of One-error using the XGBoost classifier: Dataset 14.

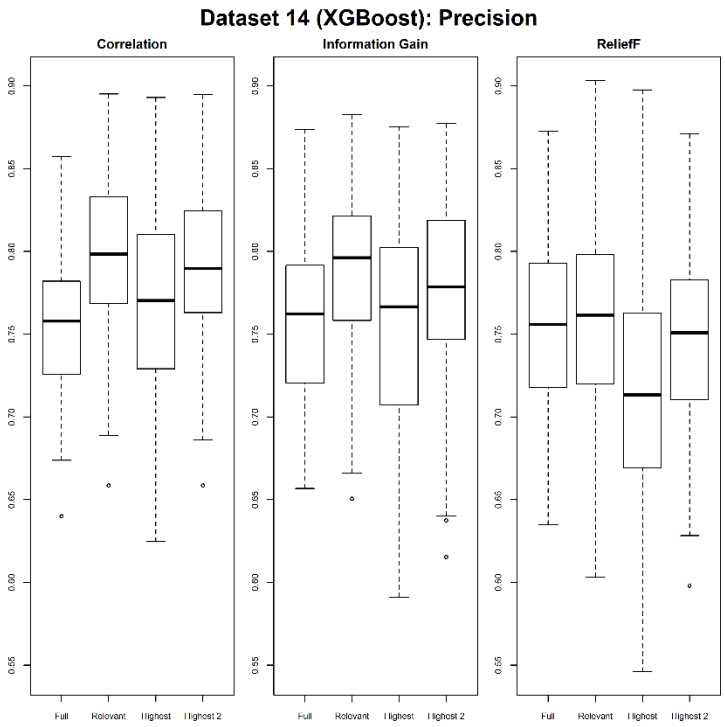


Figure C.55 Comparison of Precision using the XGBoost classifier: Dataset 14.

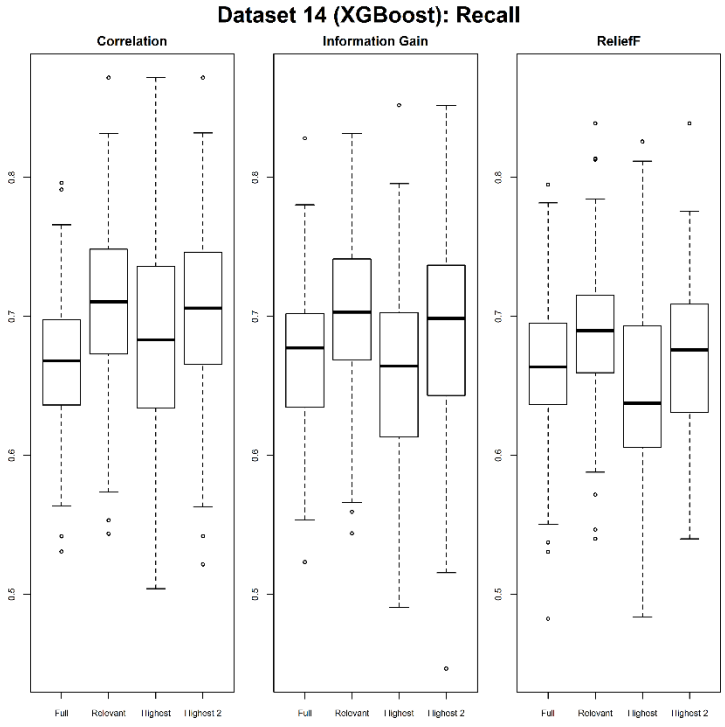


Figure C.56 Comparison of Recall using the XGBoost classifier: Dataset 14.

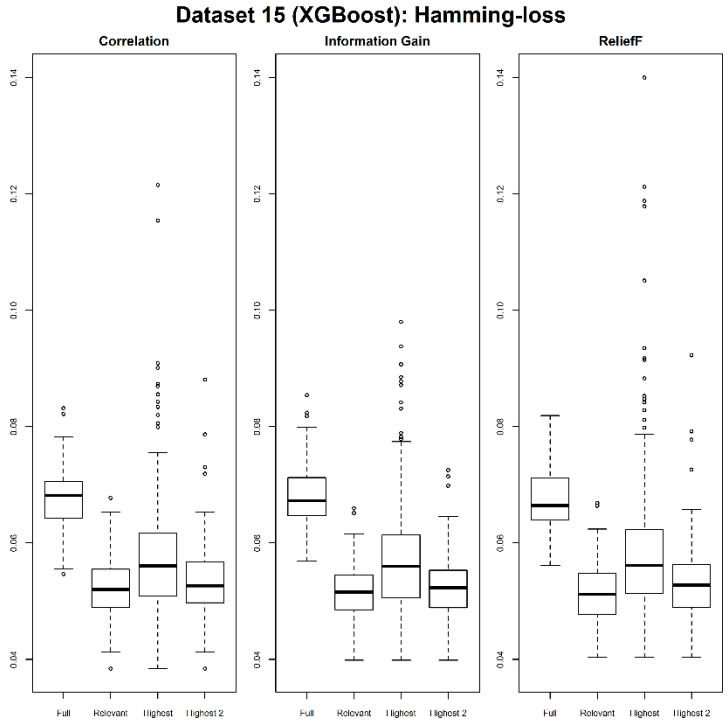


Figure C.57 Comparison of Hamming-loss using the XGBoost classifier: Dataset 15.

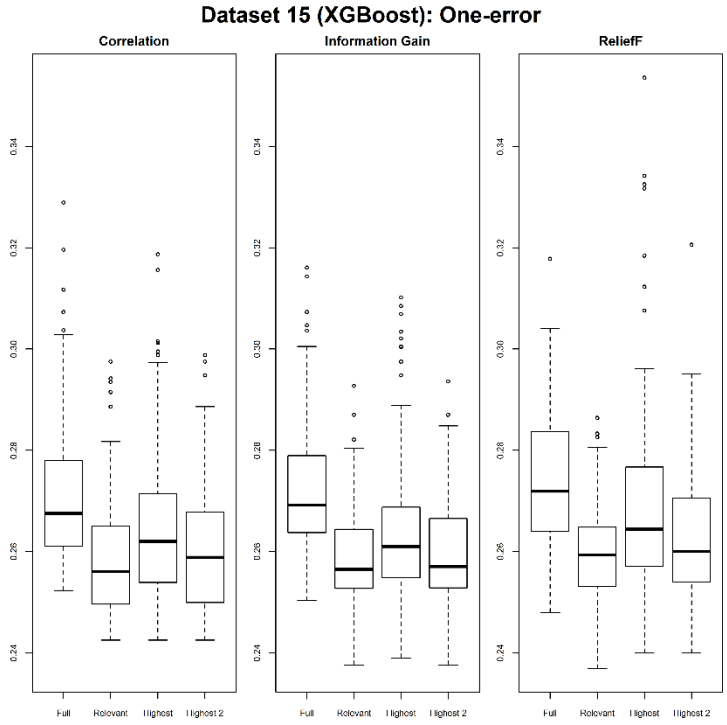


Figure C.58 Comparison of One-error using the XGBoost classifier: Dataset 15.

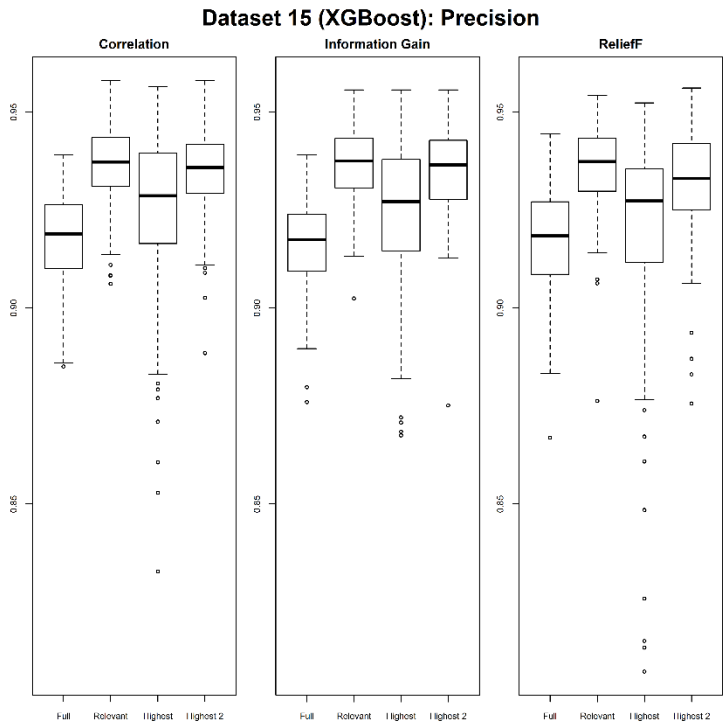


Figure C.59 Comparison of Precision using the XGBoost classifier: Dataset 15.

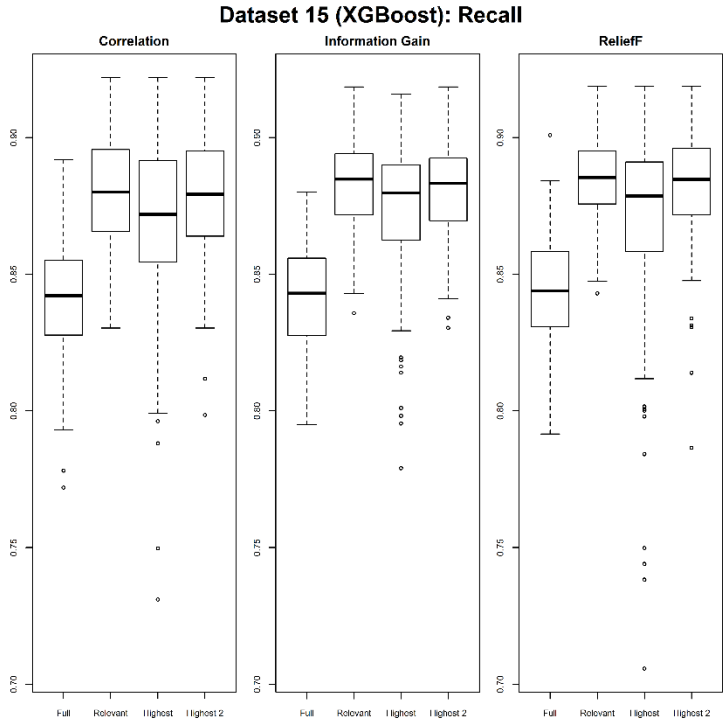


Figure C.60 Comparison of Recall using the XGBoost classifier: Dataset 15.

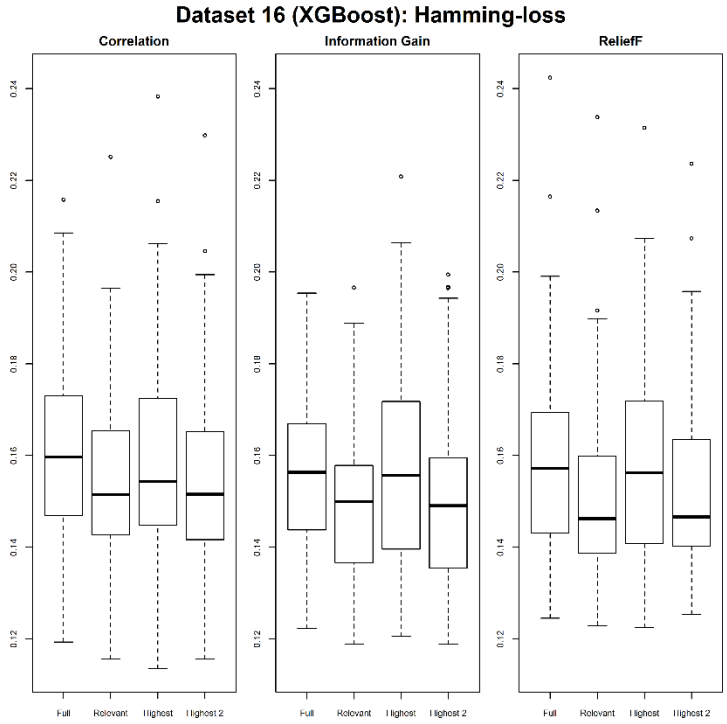


Figure C.61 Comparison of Hamming-loss using the XGBoost classifier: Dataset 16.

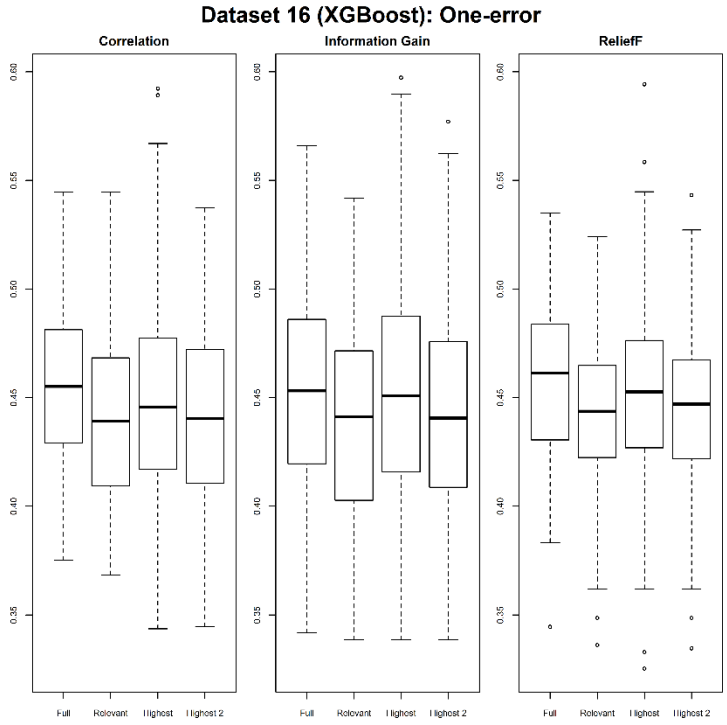


Figure C.62 Comparison of One-error using the XGBoost classifier: Dataset 16.

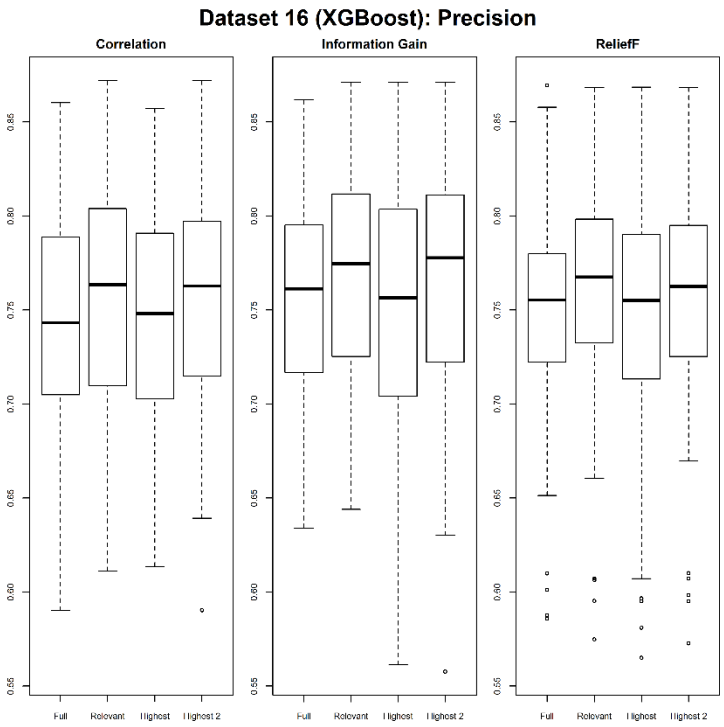


Figure C.63 Comparison of Precision using the XGBoost classifier: Dataset 16.

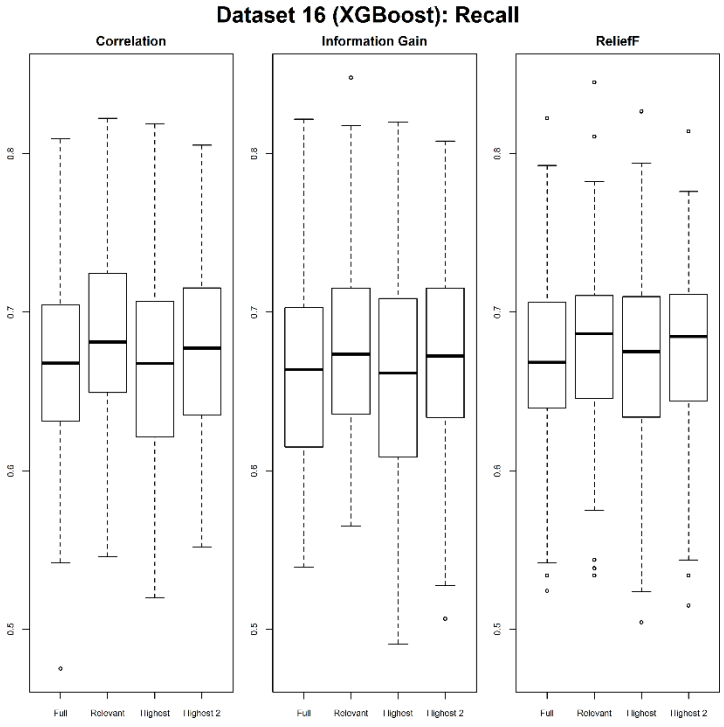


Figure C.64 Comparison of Recall using the XGBoost classifier: Dataset 16.

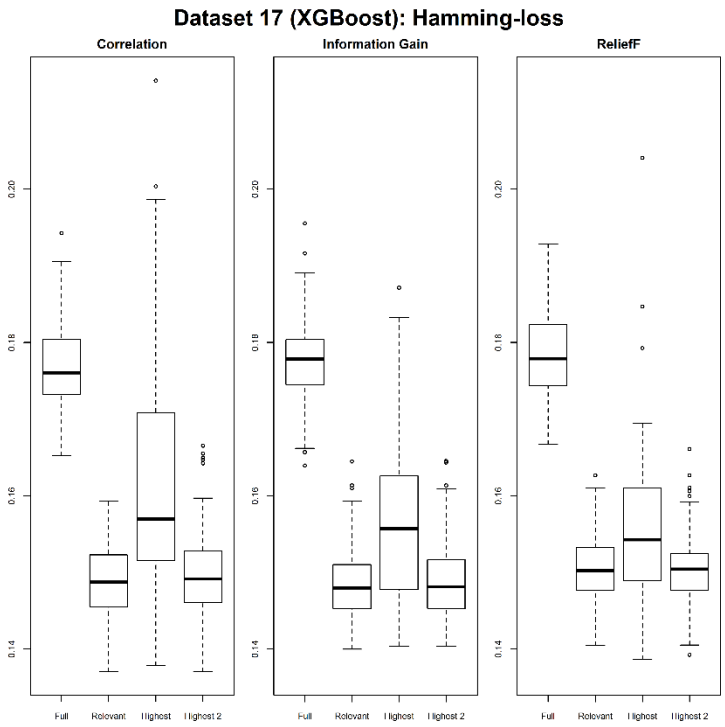


Figure C.65 Comparison of Hamming-loss using the XGBoost classifier: Dataset 17.

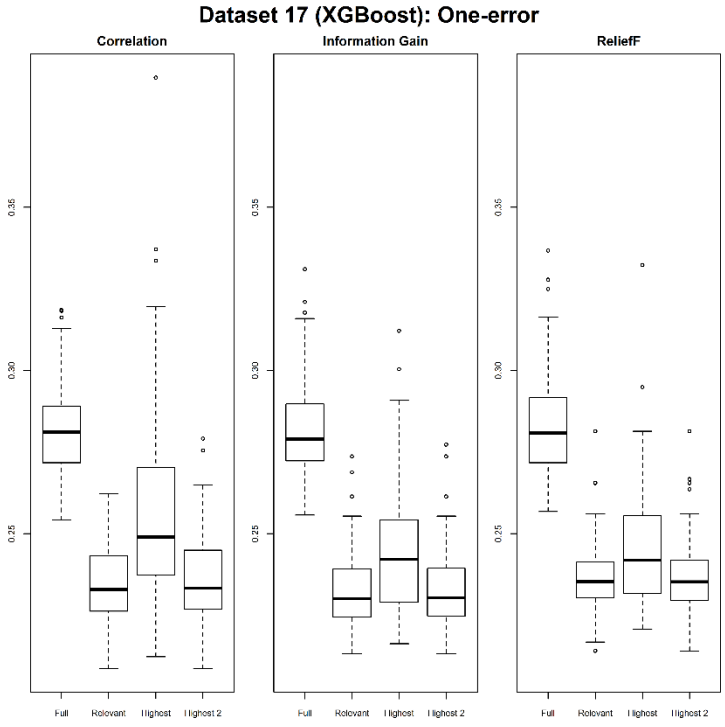


Figure C.66 Comparison of One-error using the XGBoost classifier: Dataset 17.

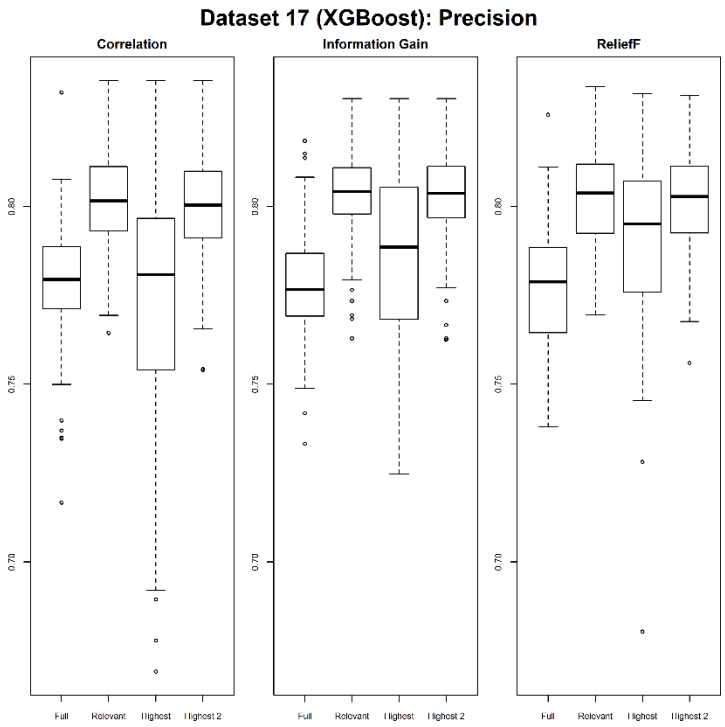


Figure C.67 Comparison of Precision using the XGBoost classifier: Dataset 17.

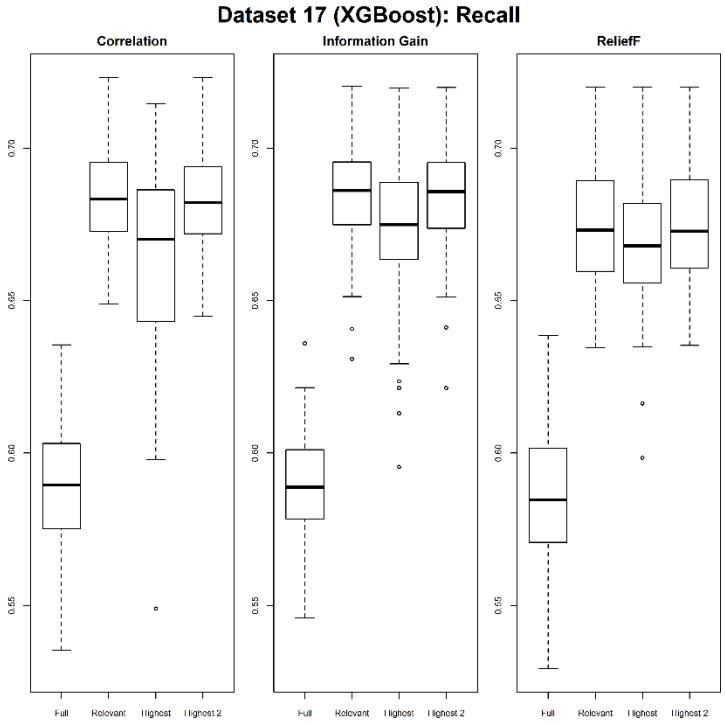


Figure C.68 Comparison of Recall using the XGBoost classifier: Dataset 17.

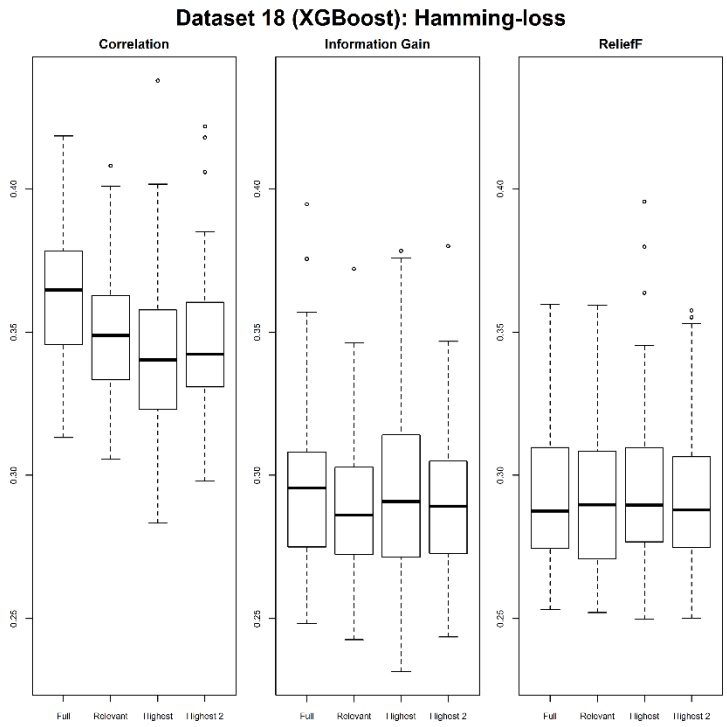


Figure C.69 Comparison of Hamming-loss using the XGBoost classifier: Dataset 18.

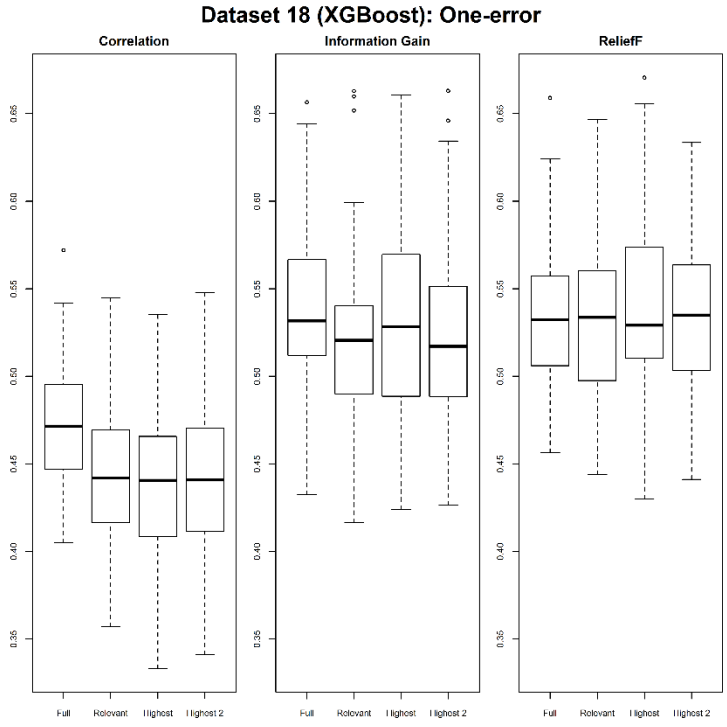


Figure C.70 Comparison of One-error using the XGBoost classifier: Dataset 18.

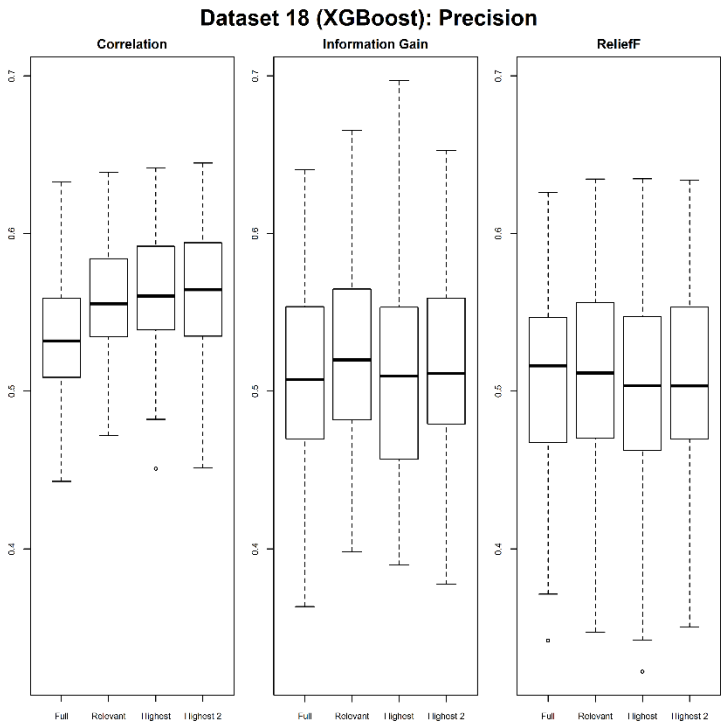


Figure C.71 Comparison of Precision using the XGBoost classifier: Dataset 18.

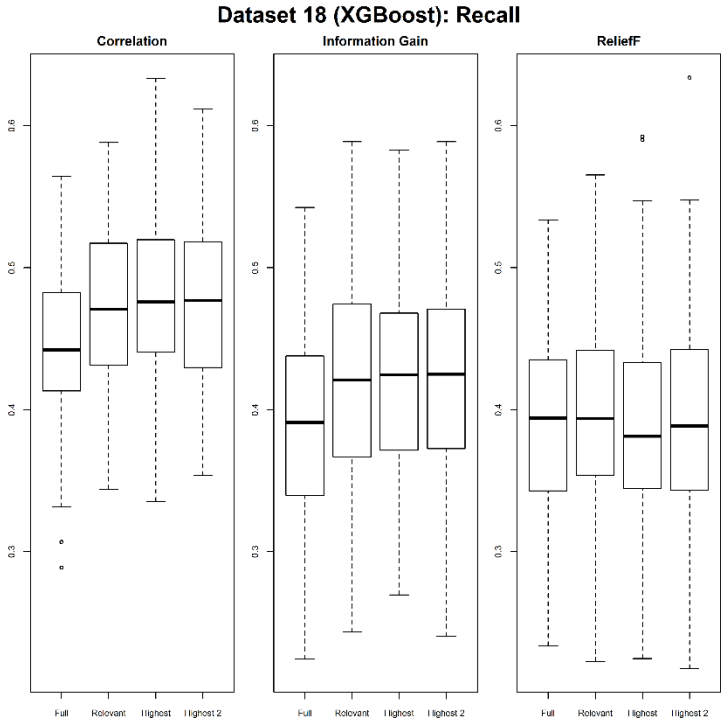


Figure C.72 Comparison of Recall using the XGBoost classifier: Dataset 18.

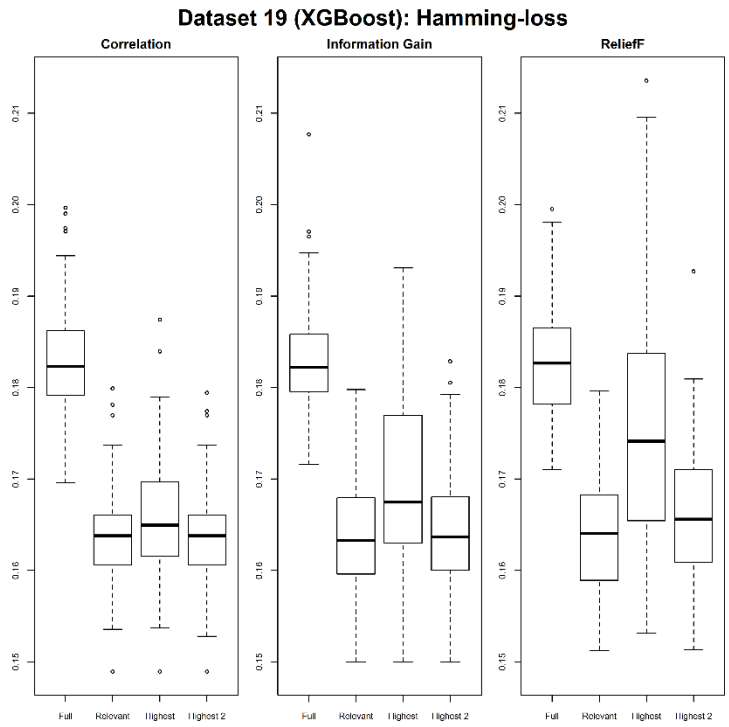


Figure C.73 Comparison of Hamming-loss using the XGBoost classifier: Dataset 19.

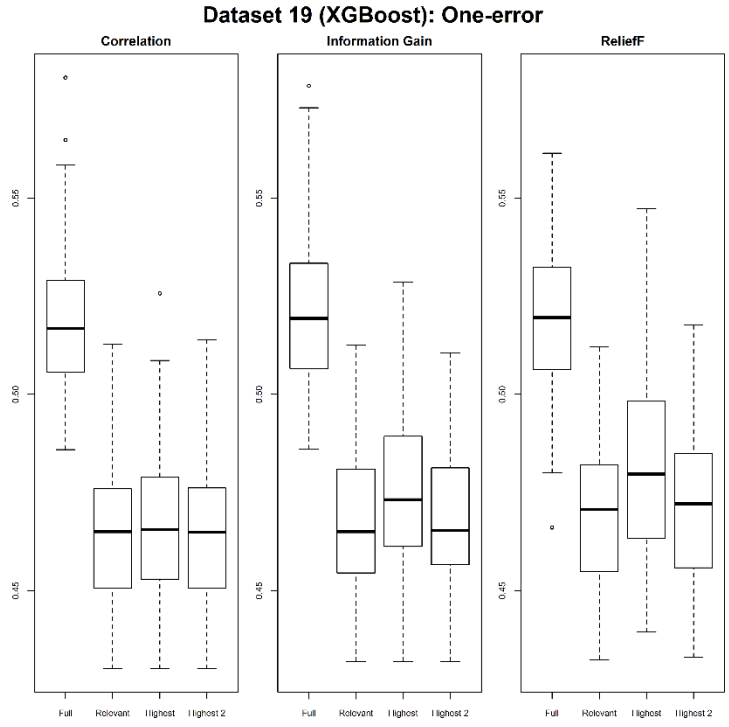


Figure C.74 Comparison of One-error using the XGBoost classifier: Dataset 19.

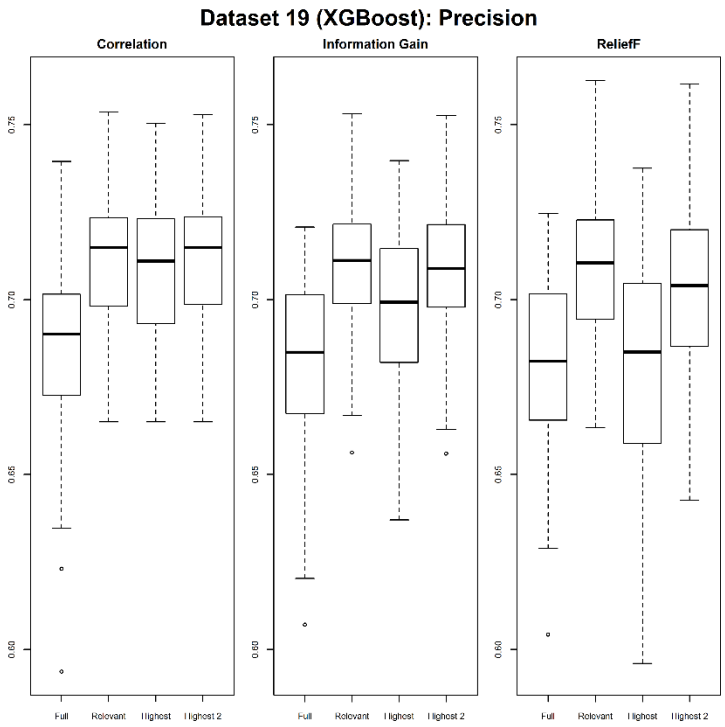


Figure C.75 Comparison of Precision using the XGBoost classifier: Dataset 19.

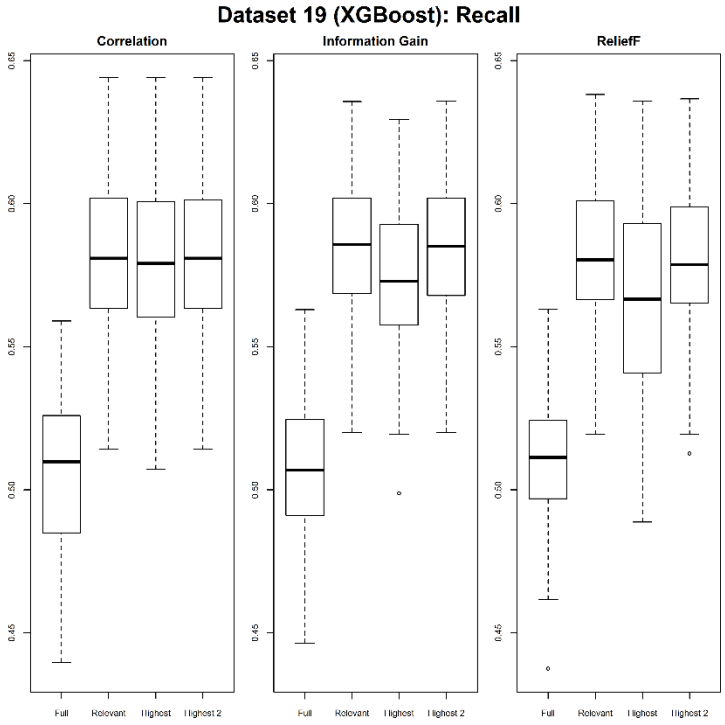


Figure C.76 Comparison of Recall using the XGBoost classifier: Dataset 19.

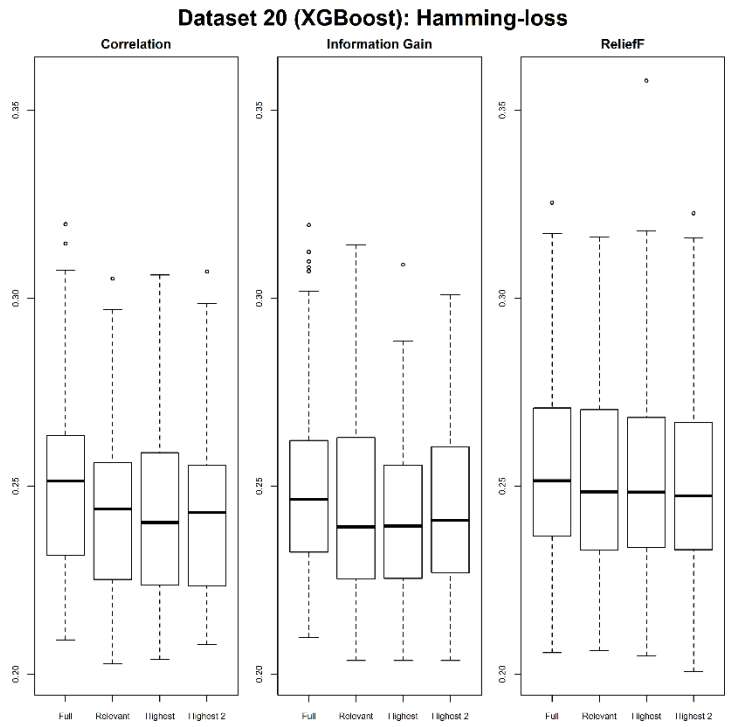


Figure C.77 Comparison of Hamming-loss using the XGBoost classifier: Dataset 20.

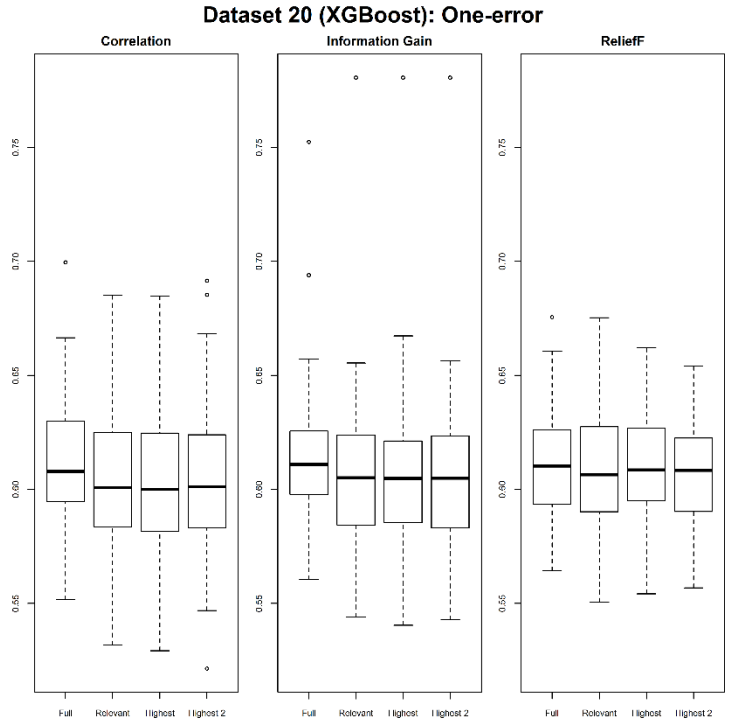


Figure C.78 Comparison of One-error using the XGBoost classifier: Dataset 20.

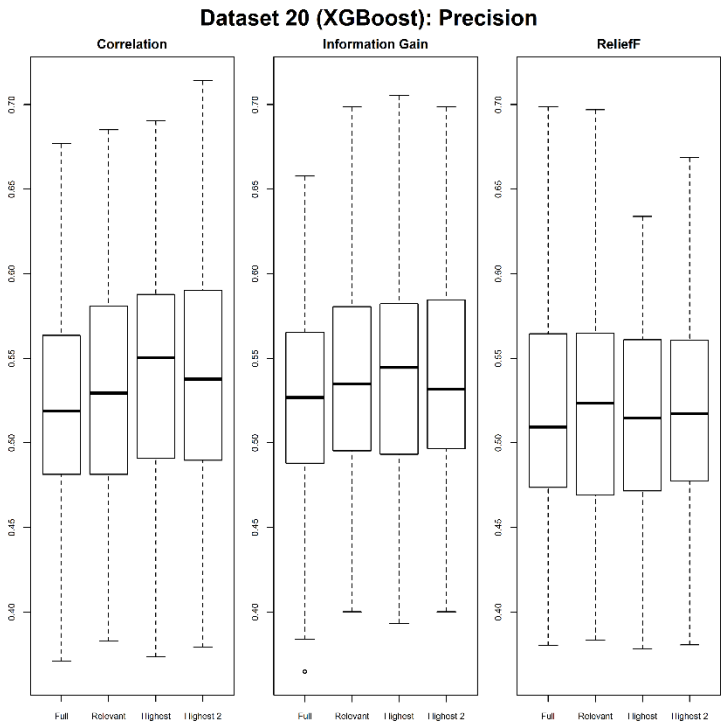


Figure C.79 Comparison of Precision using the XGBoost classifier: Dataset 20.

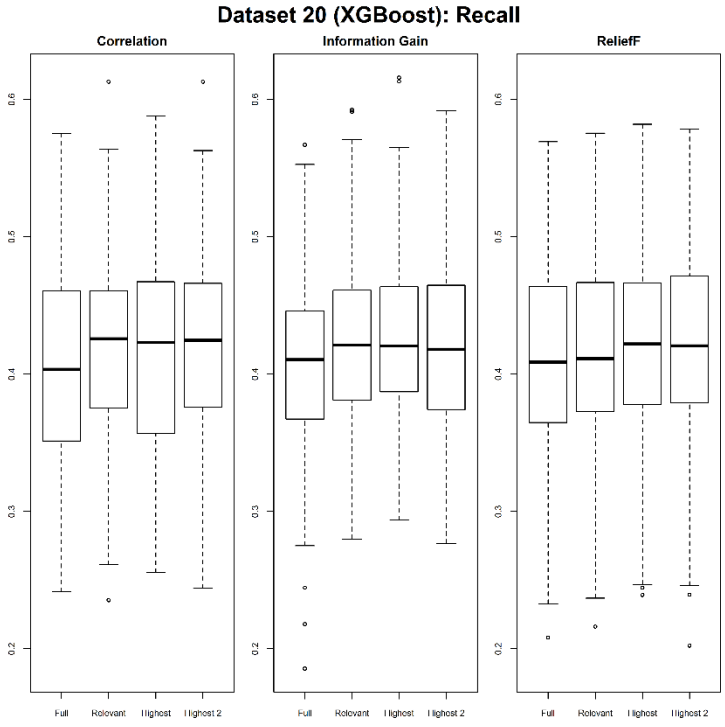


Figure C.80 Comparison of Recall using the XGBoost classifier: Dataset 20.

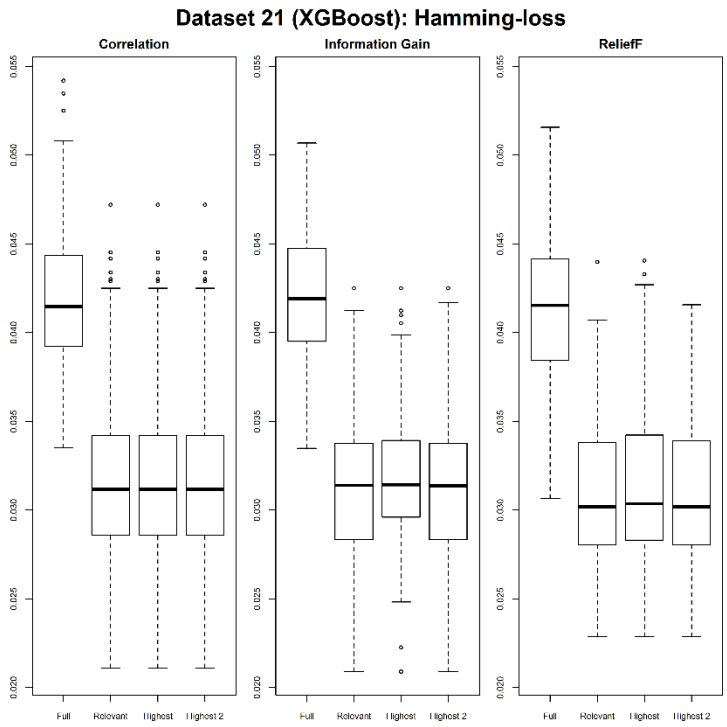


Figure C.81 Comparison of Hamming-loss using the XGBoost classifier: Dataset 21.

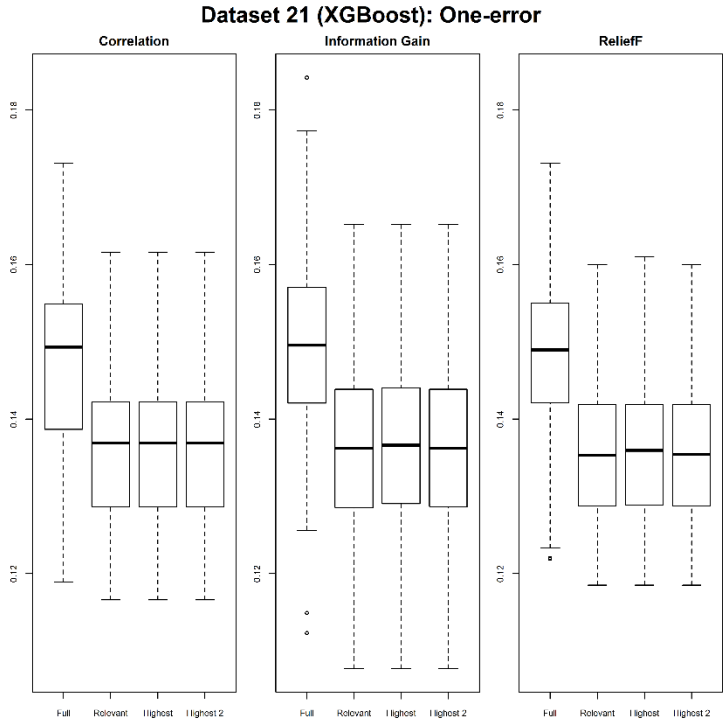


Figure C.82 Comparison of One-error using the XGBoost classifier: Dataset 21.

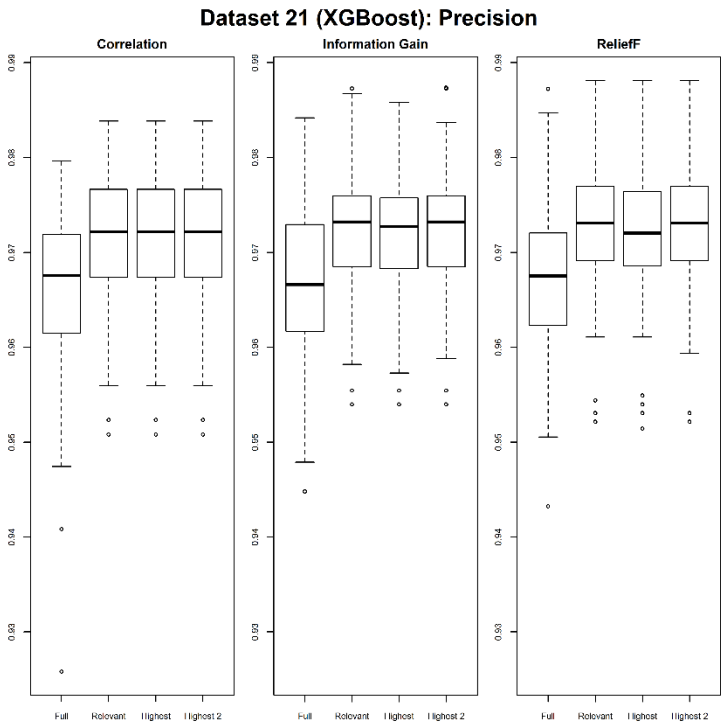


Figure C.83 Comparison of Precision using the XGBoost classifier: Dataset 21.

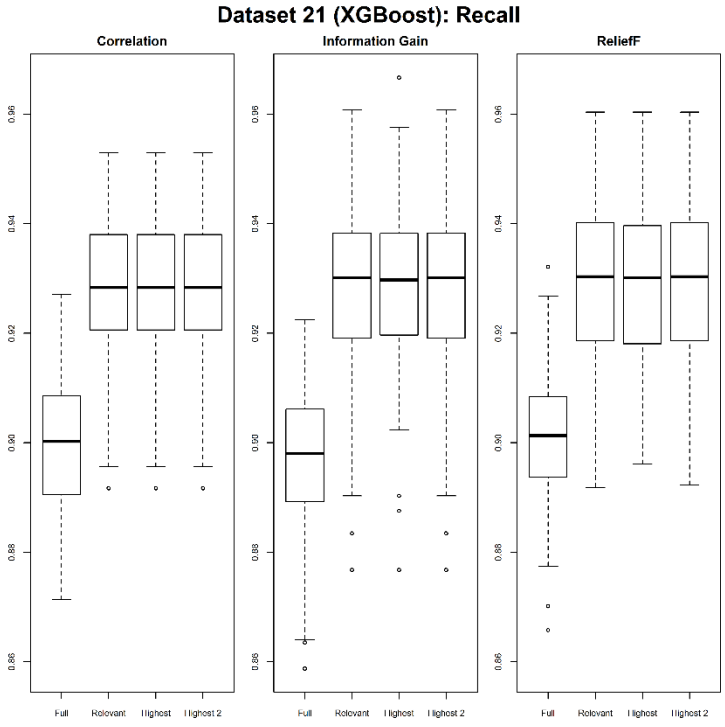


Figure C.84 Comparison of Recall using the XGBoost classifier: Dataset 21.

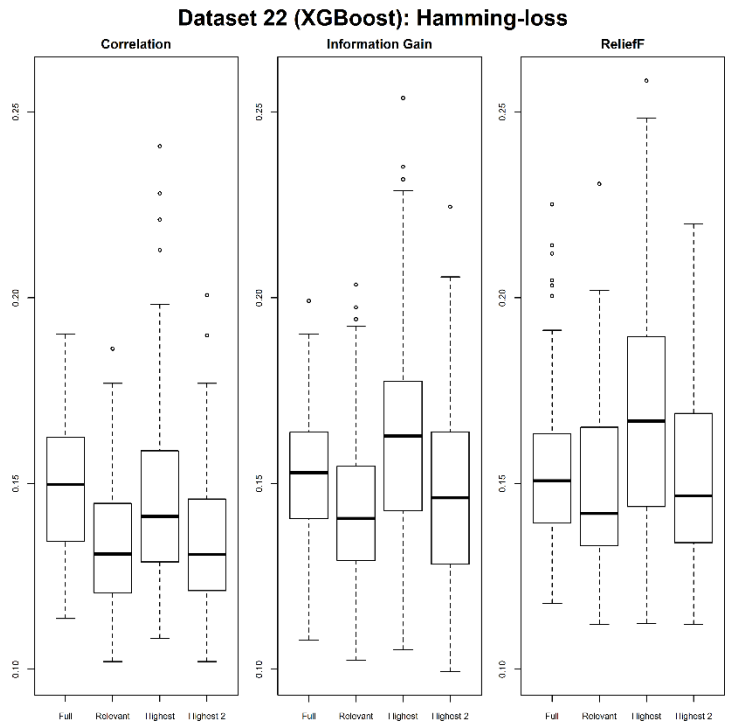


Figure C.85 Comparison of Hamming-loss using the XGBoost classifier: Dataset 22.

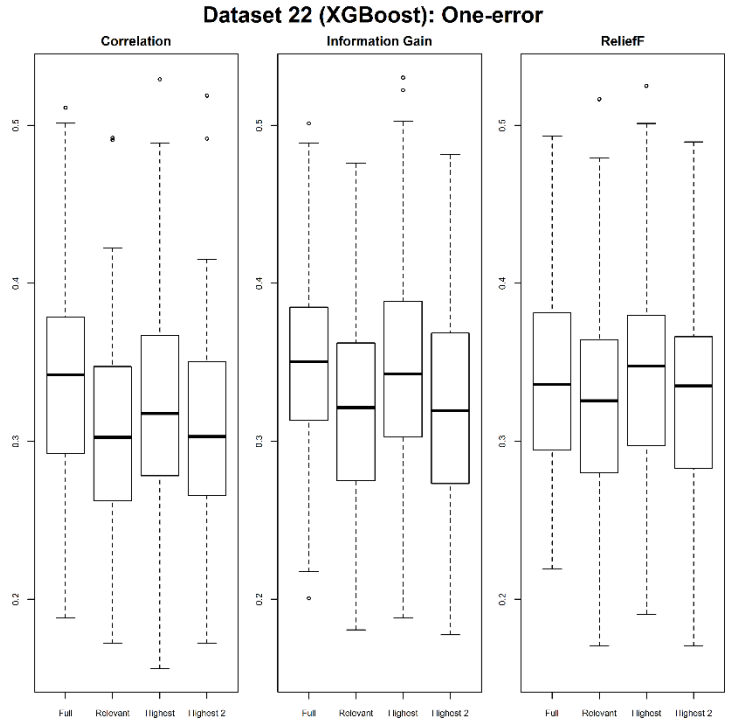


Figure C.86 Comparison of One-error using the XGBoost classifier: Dataset 22.

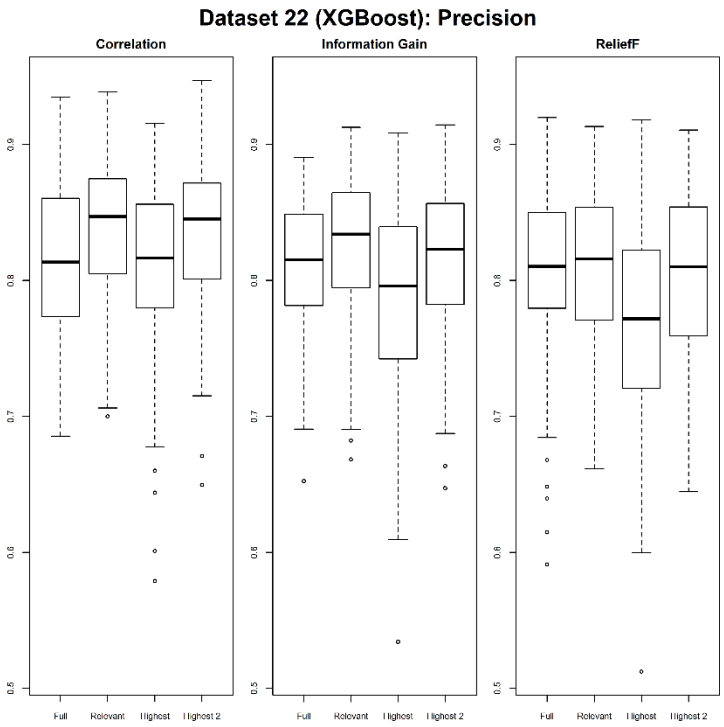


Figure C.87 Comparison of Precision using the XGBoost classifier: Dataset 22.

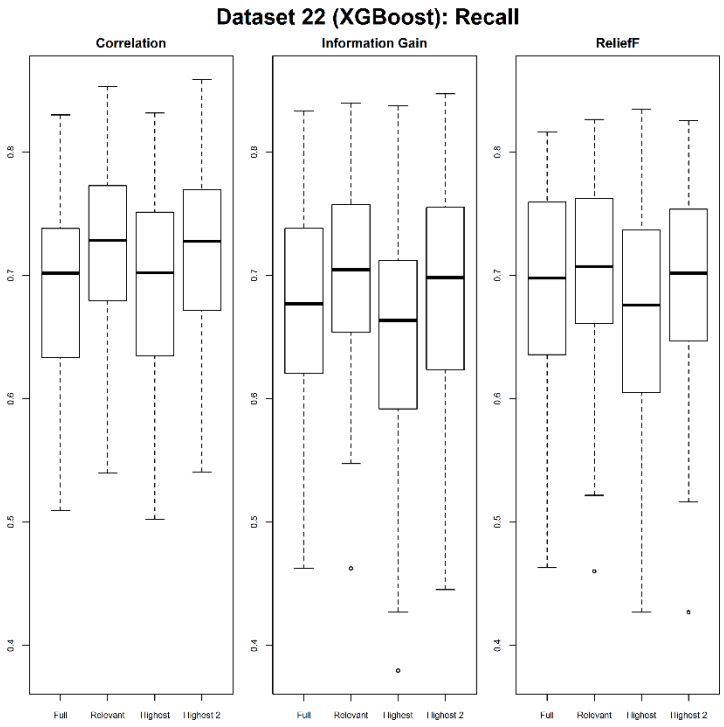


Figure C.88 Comparison of Recall using the XGBoost classifier: Dataset 22.

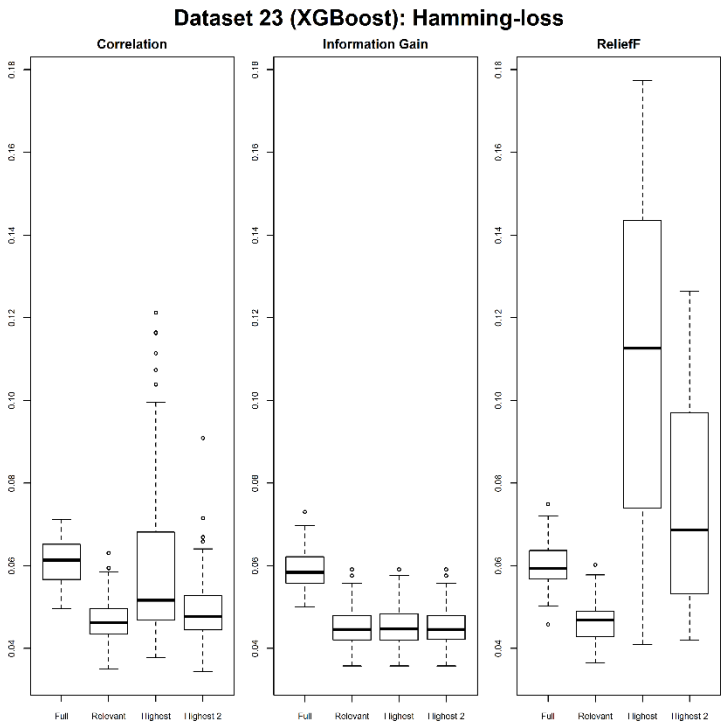


Figure C.89 Comparison of Hamming-loss using the XGBoost classifier: Dataset 23.

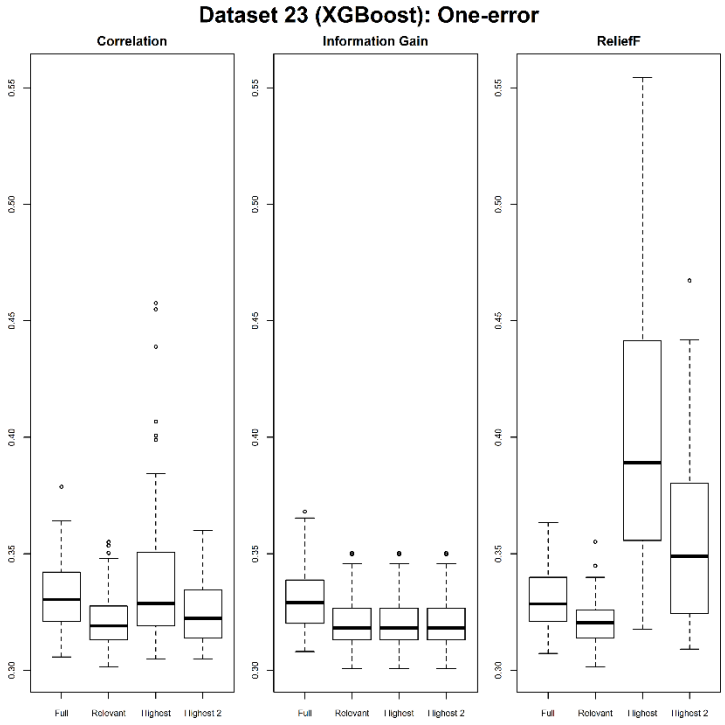


Figure C.90 Comparison of One-error using the XGBoost classifier: Dataset 23.

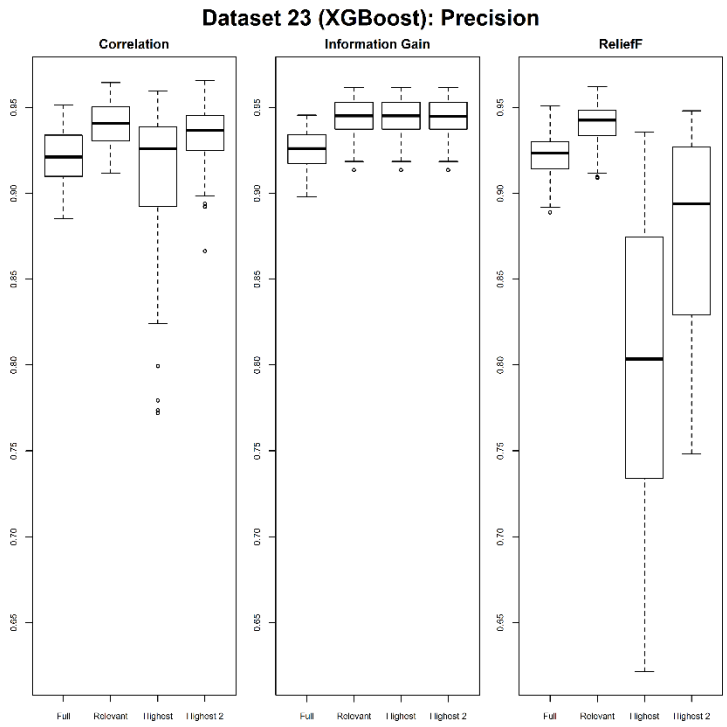


Figure C.91 Comparison of Precision using the XGBoost classifier: Dataset 23.

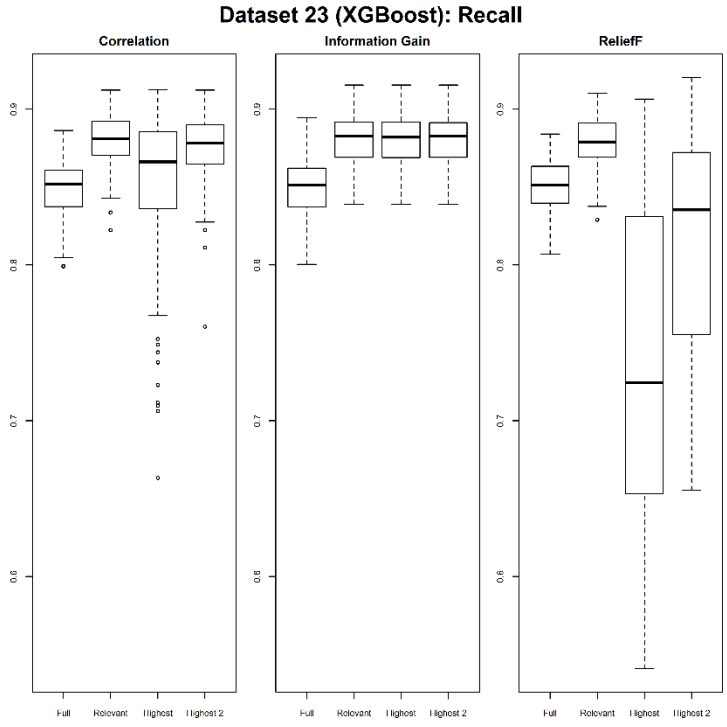


Figure C.92 Comparison of Recall using the XGBoost classifier: Dataset 23.

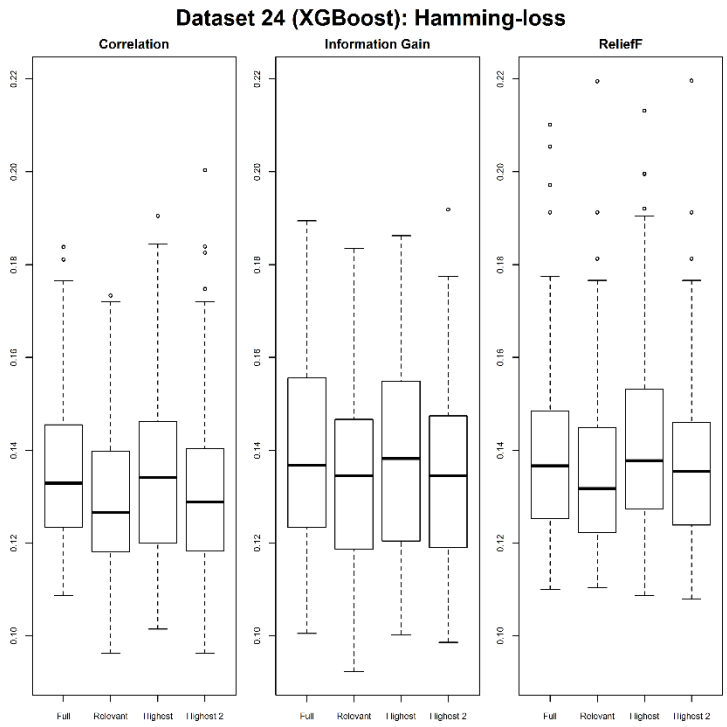


Figure C.93 Comparison of Hamming-loss using the XGBoost classifier: Dataset 24.

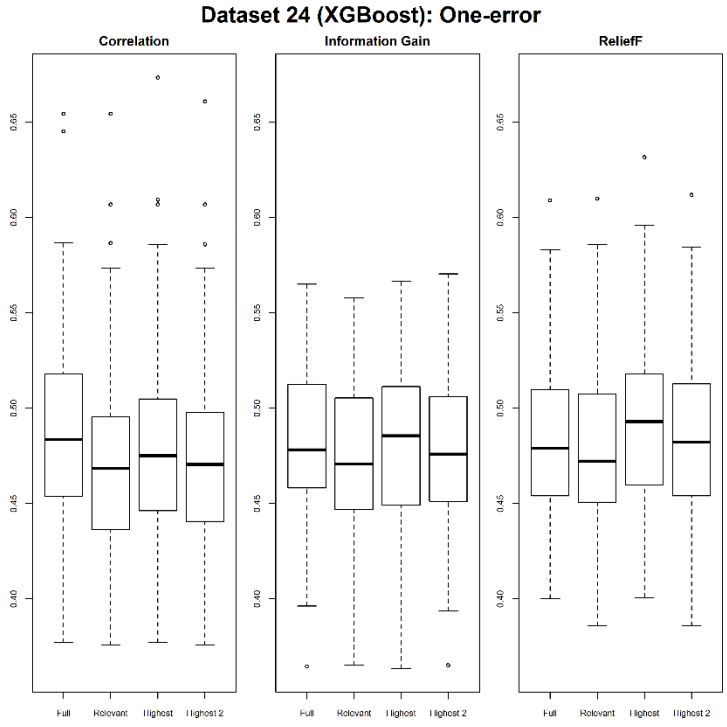


Figure C.94 Comparison of One-error using the XGBoost classifier: Dataset 24.

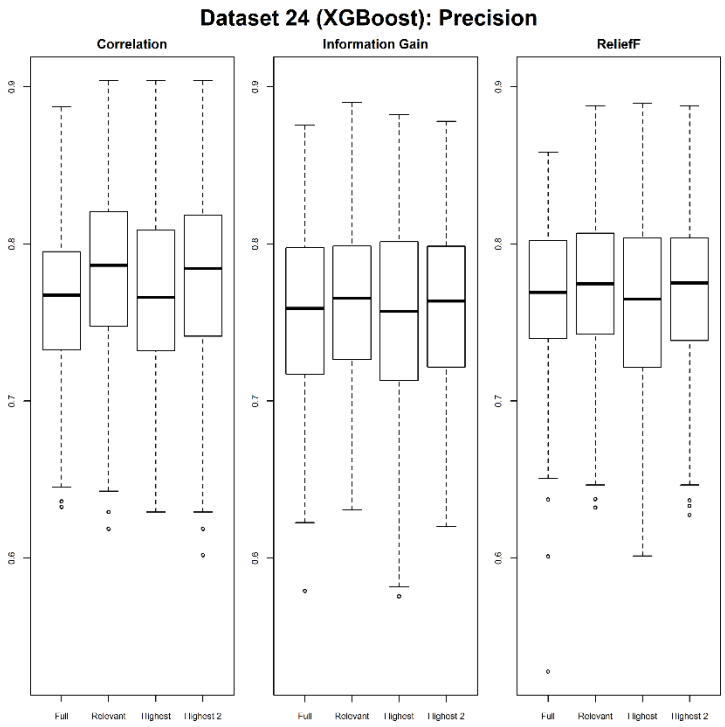


Figure C.95 Comparison of Precision using the XGBoost classifier: Dataset 24.

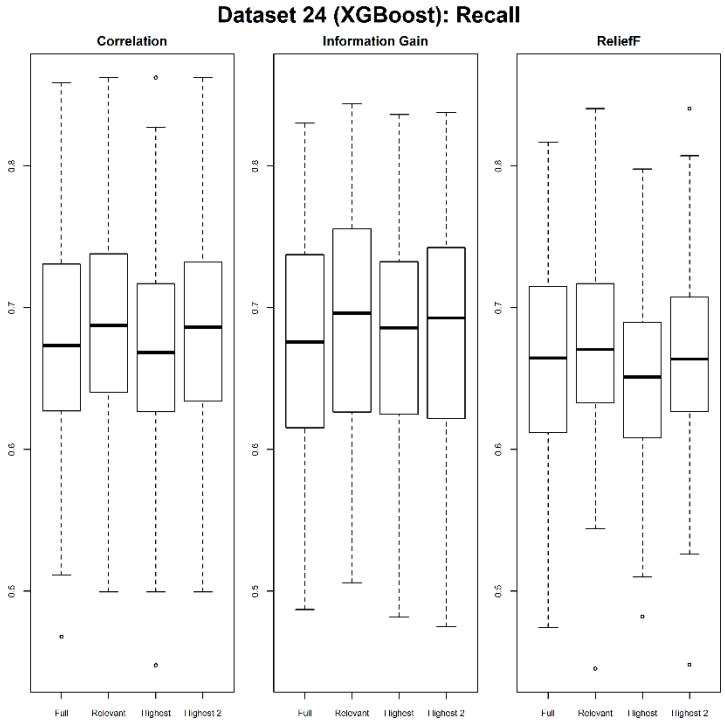


Figure C.96 Comparison of Recall using the XGBoost classifier: Dataset 24.

APPENDIX D

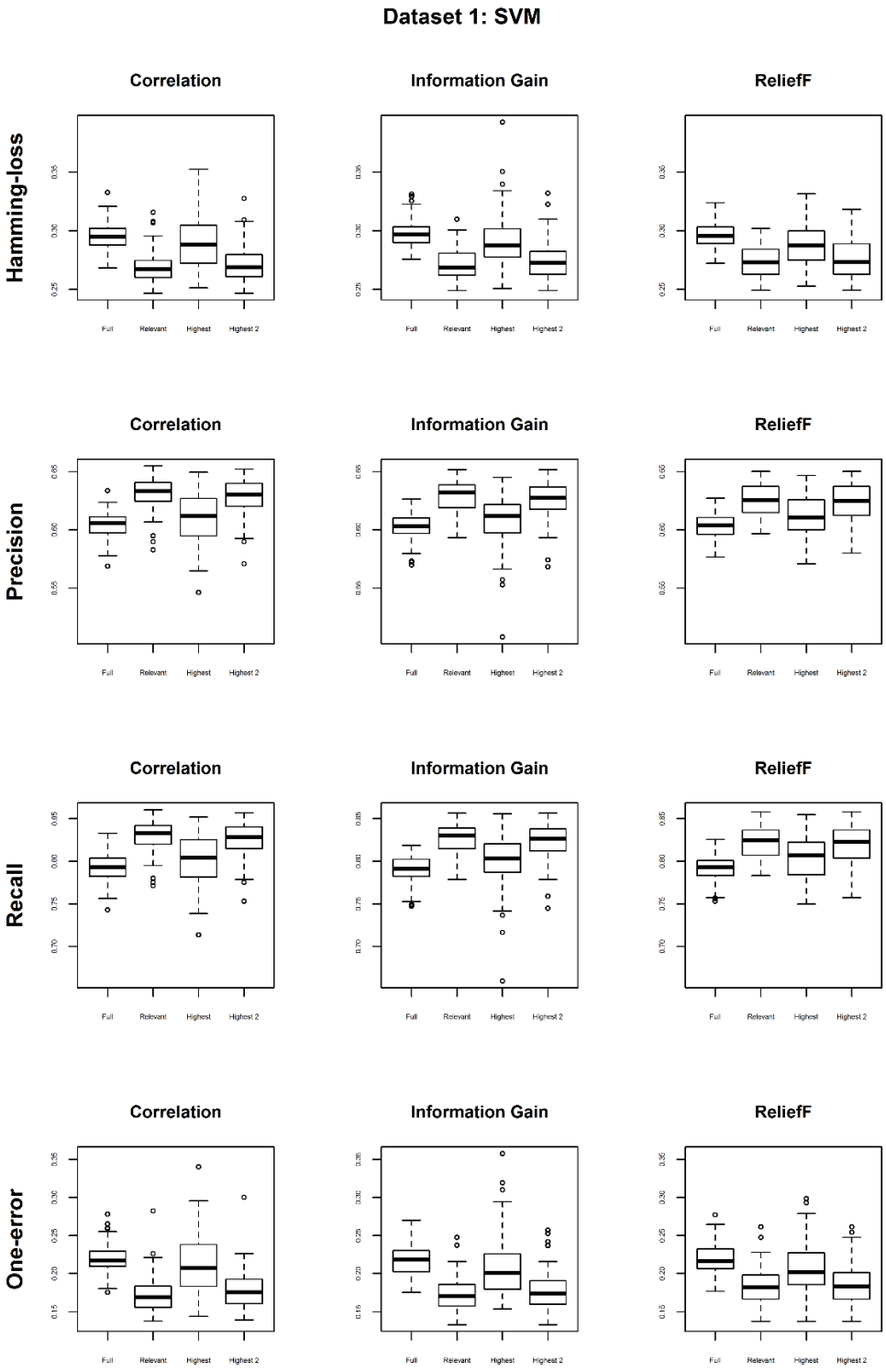


Figure D.1 Summary of results for the SVM classifier: Dataset 1.

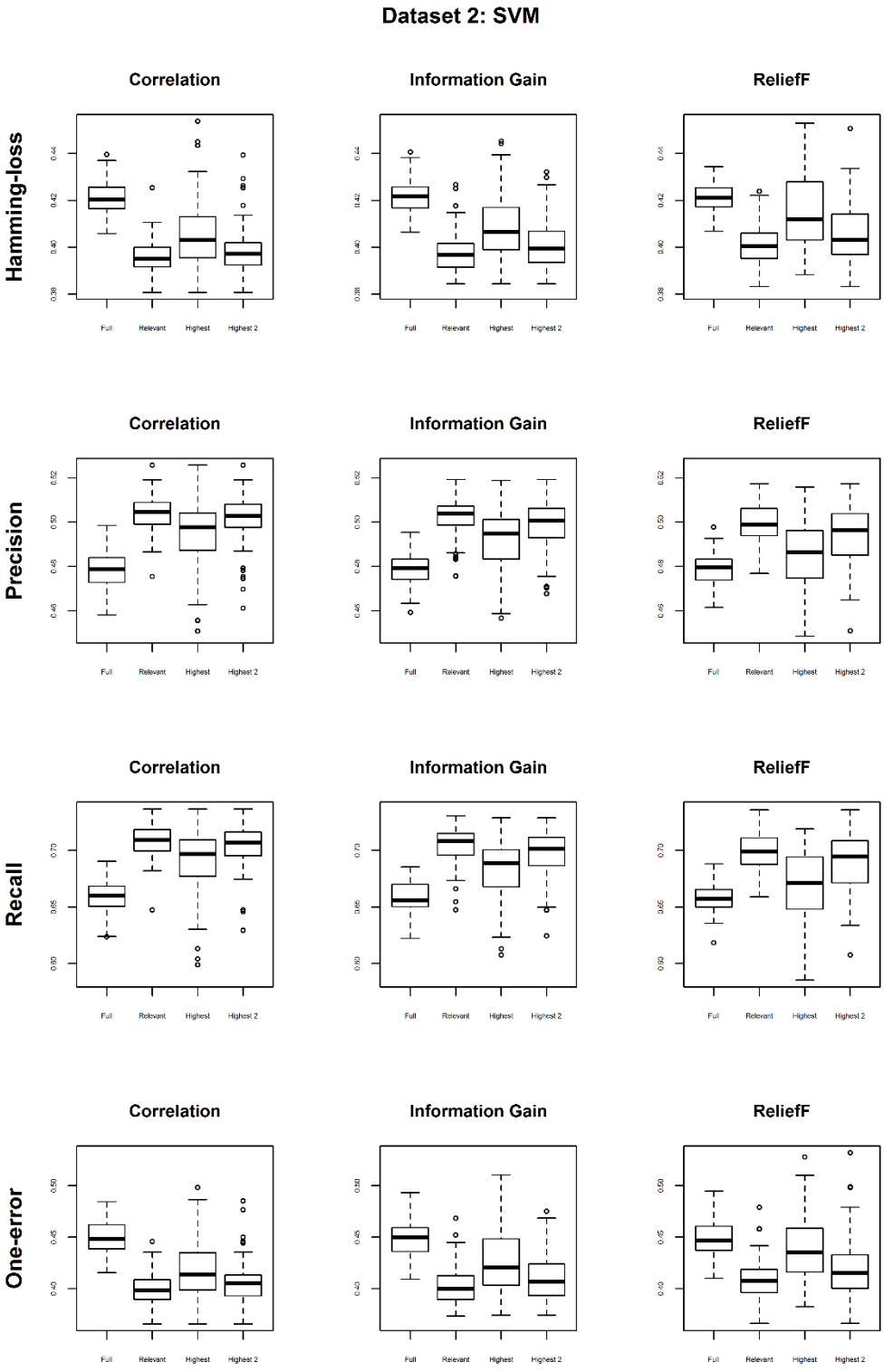


Figure D.2 Summary of results for the SVM classifier: Dataset 2.

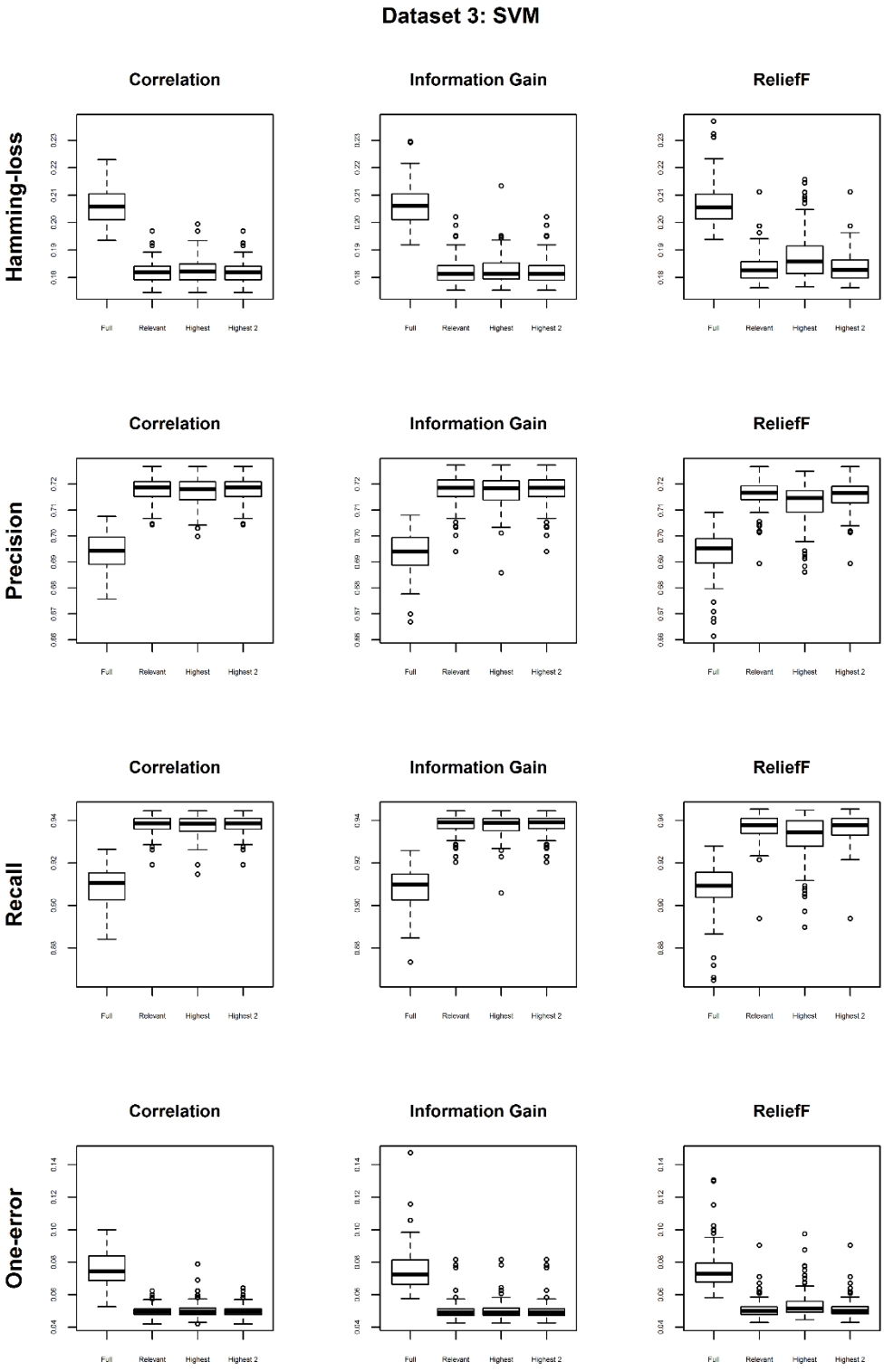


Figure D.3 Summary of results for the SVM classifier: Dataset 3.

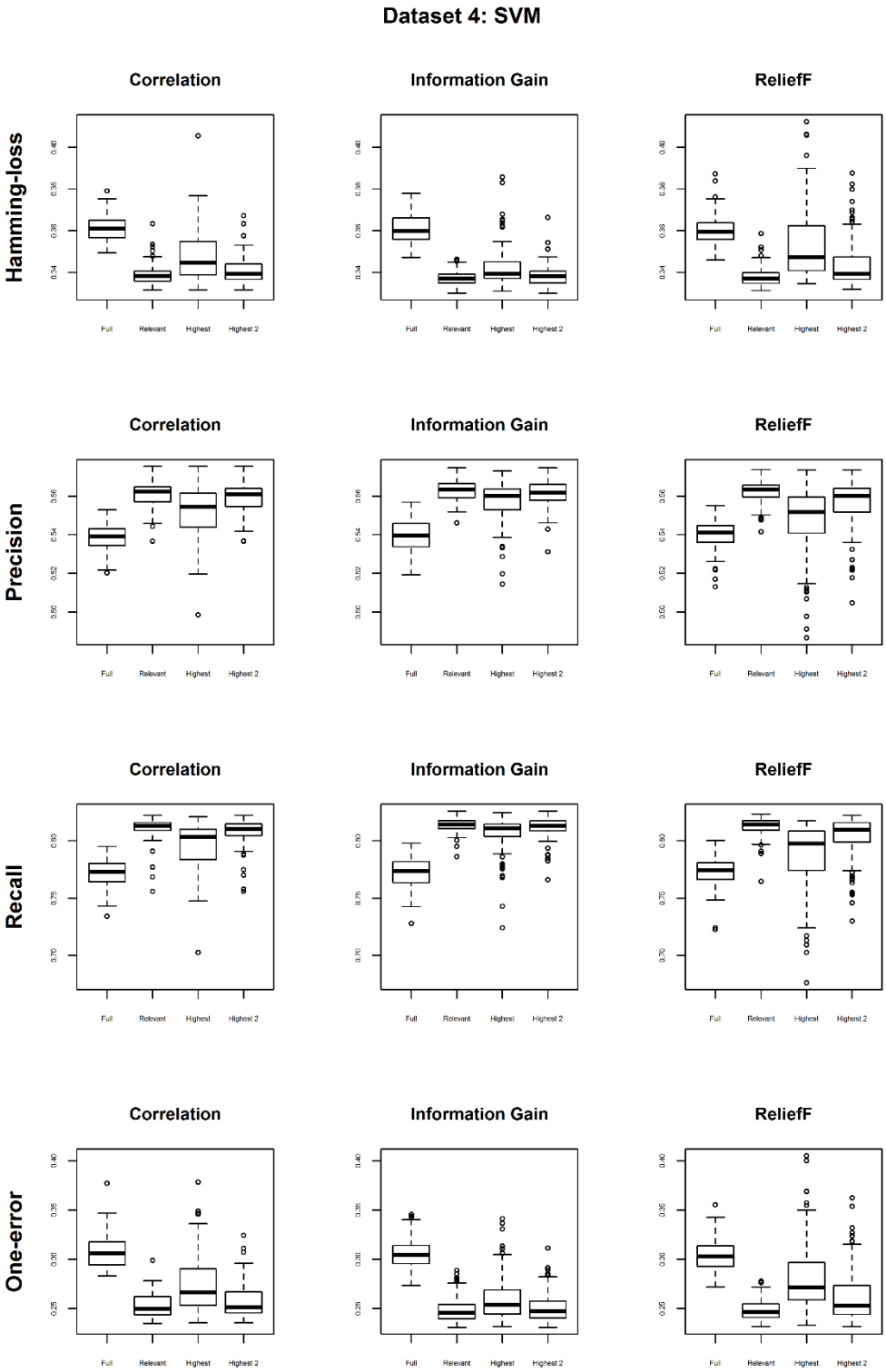


Figure D.4 Summary of results for the SVM classifier: Dataset 4.

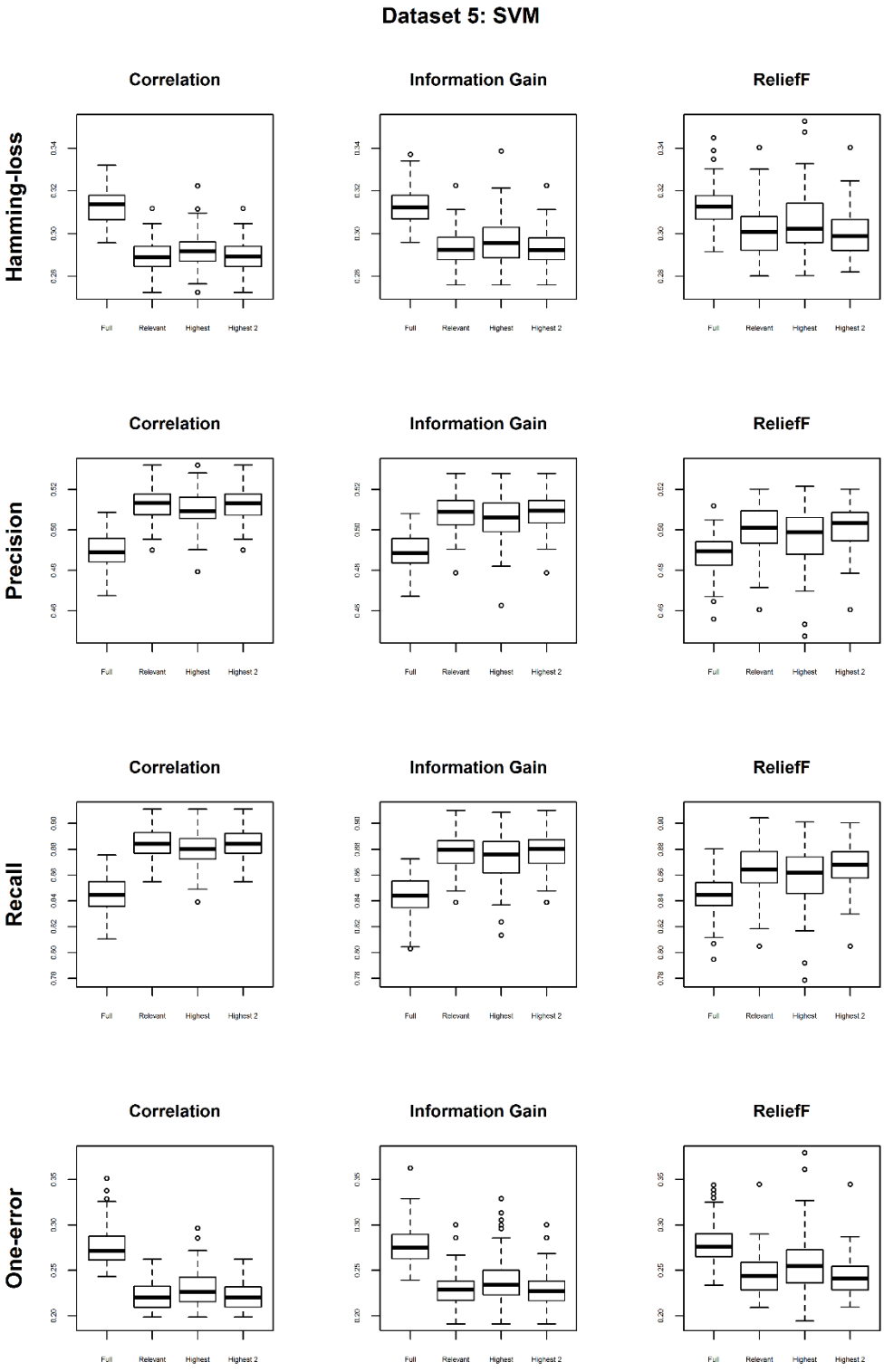


Figure D.5 Summary of results for the SVM classifier: Dataset 5.

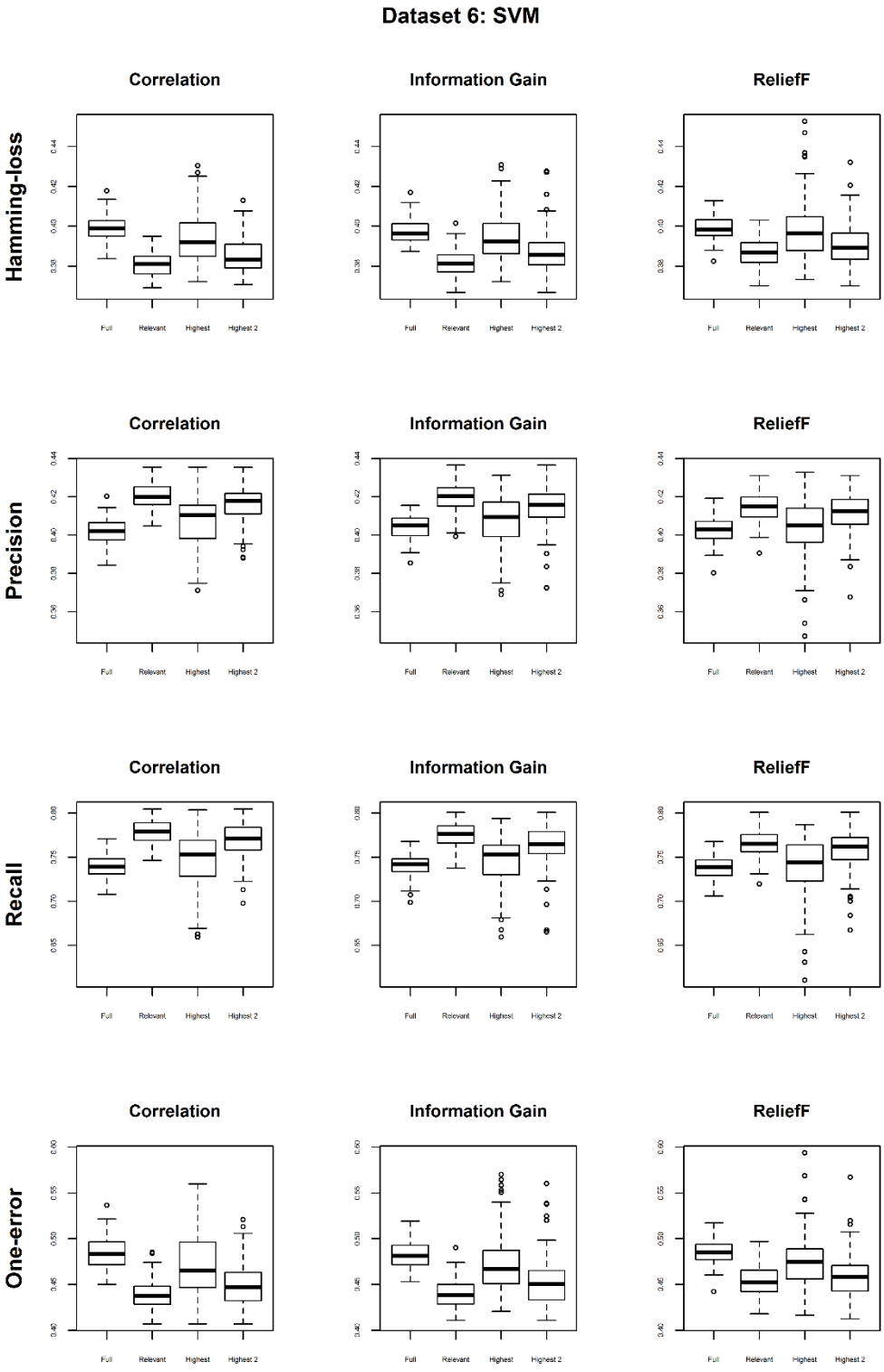


Figure D.6 Summary of results for the SVM classifier: Dataset 6.

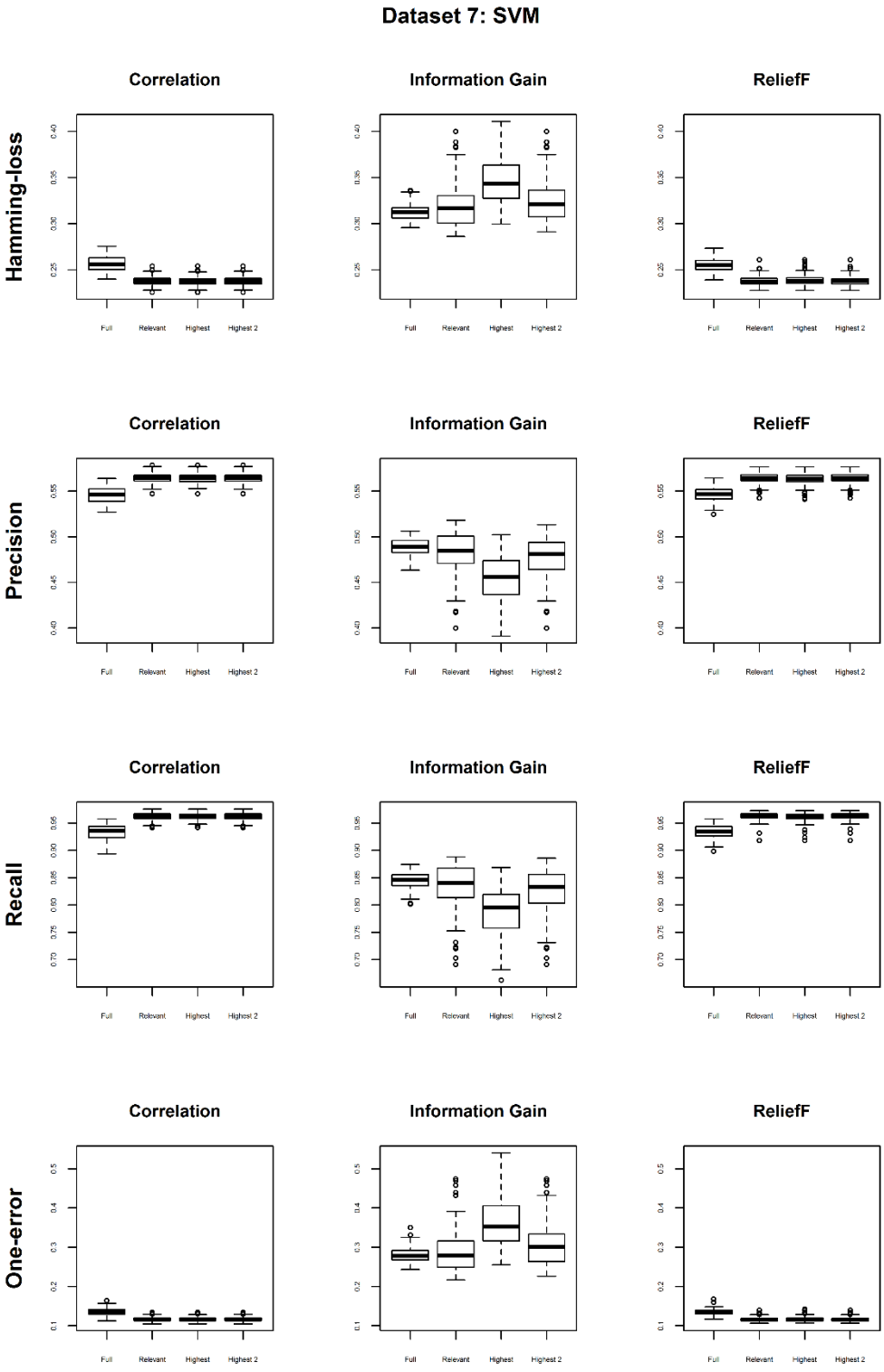


Figure D.7 Summary of results for the SVM classifier: Dataset 7.

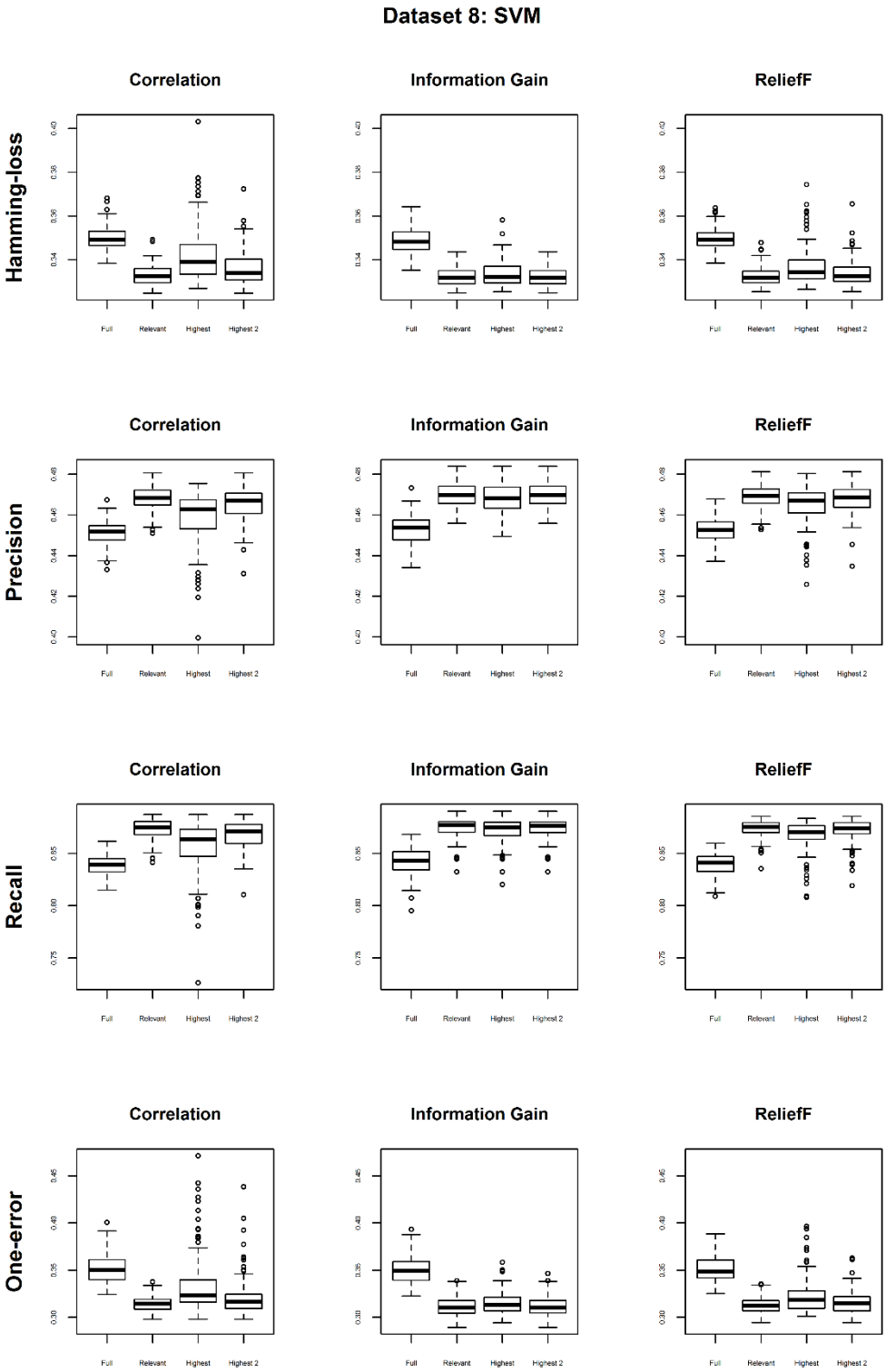


Figure D.8 Summary of results for the SVM classifier: Dataset 8.

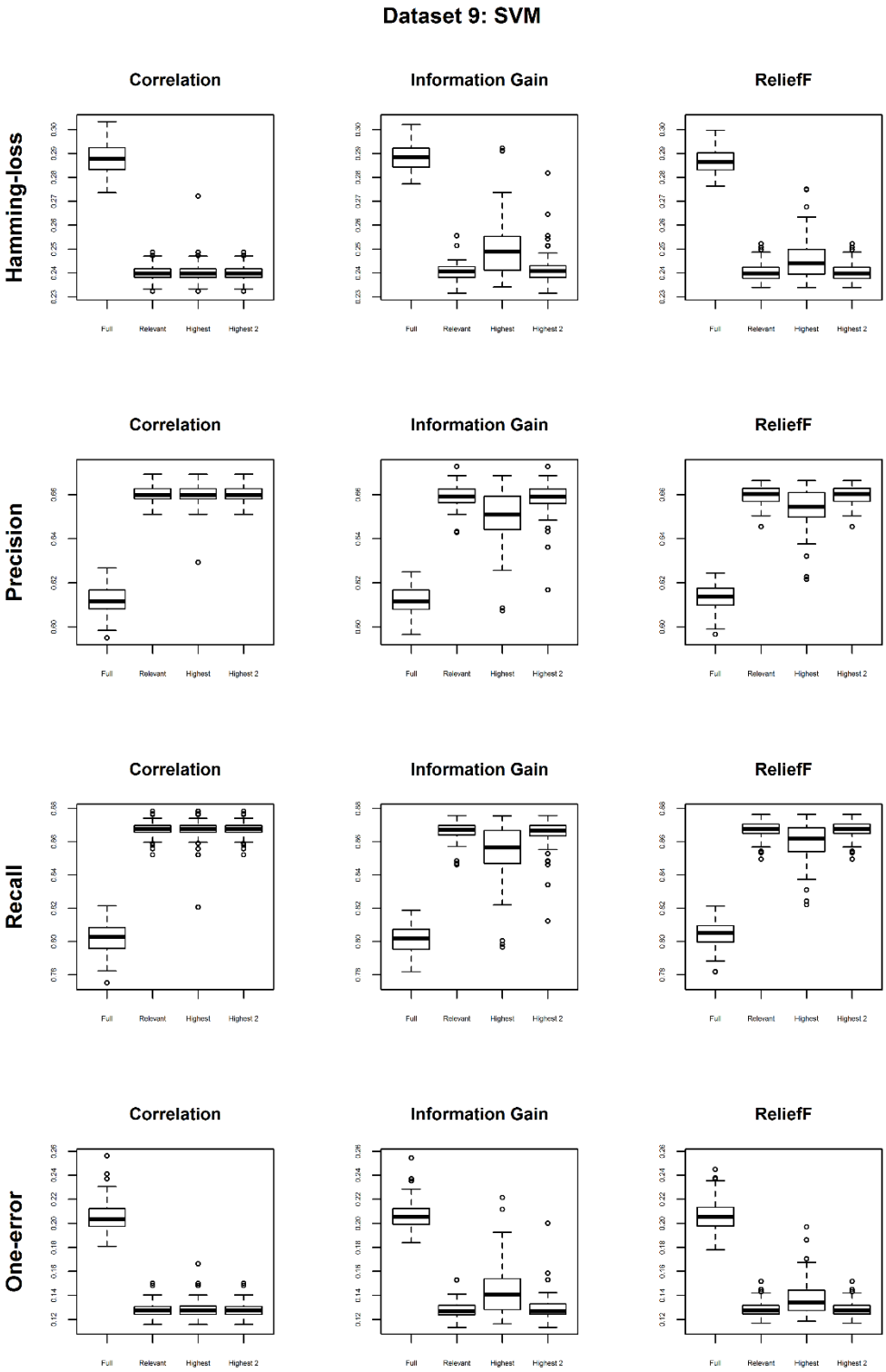


Figure D.9 Summary of results for the SVM classifier: Dataset 9.

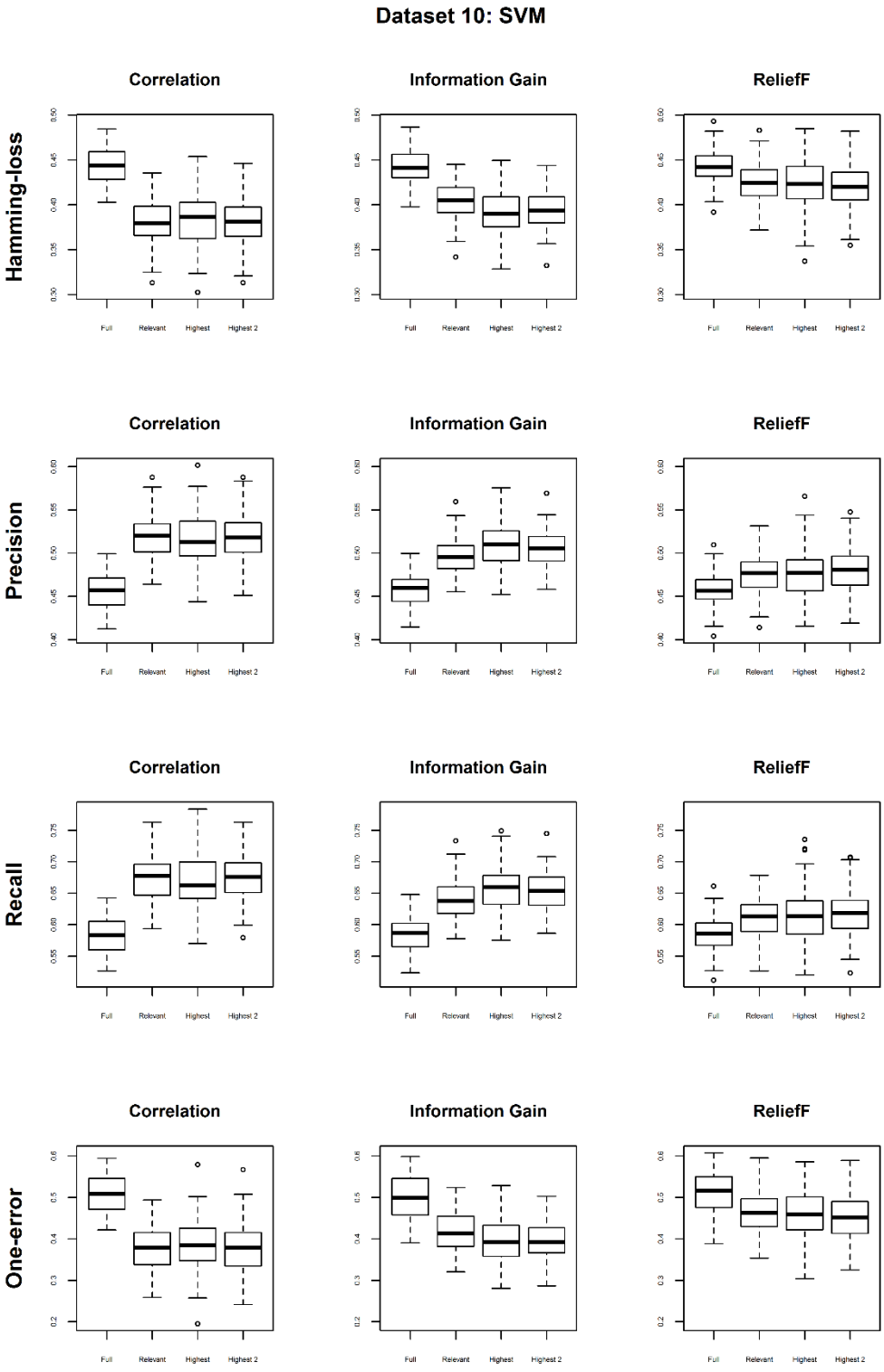


Figure D.10 Summary of results for the SVM classifier: Dataset 10.

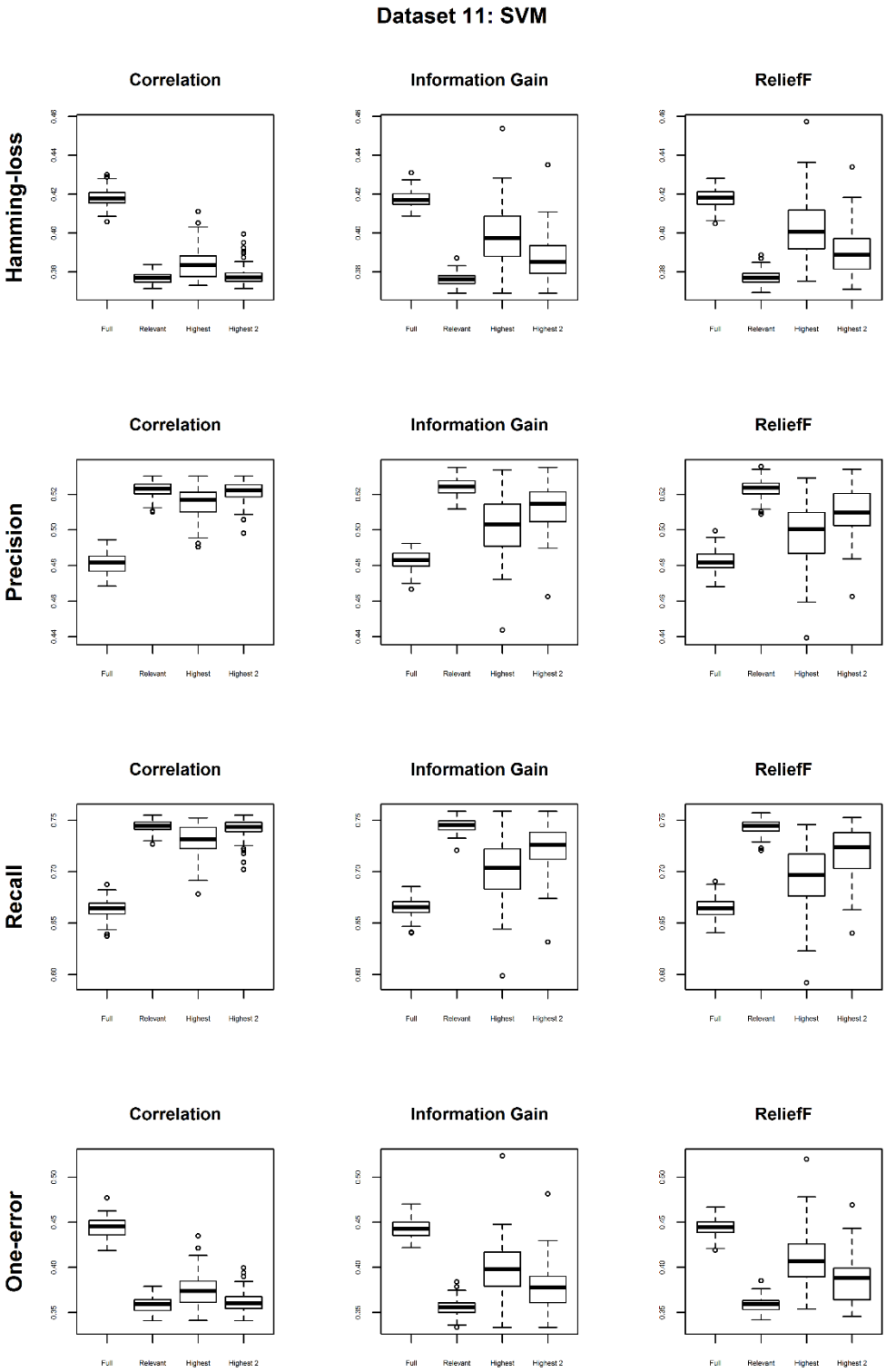


Figure D.11 Summary of results for the SVM classifier: Dataset 11.

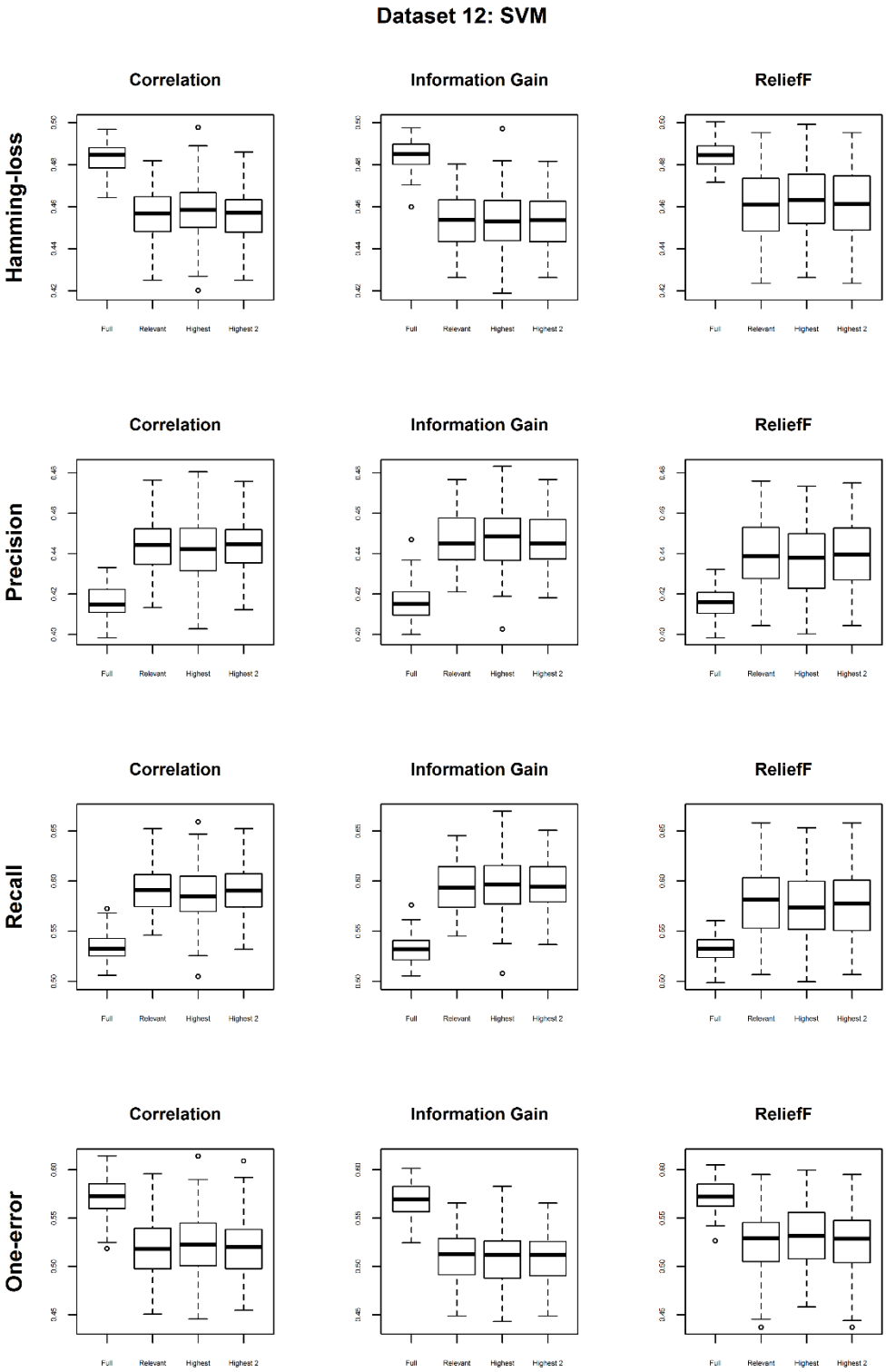


Figure D.12 Summary of results for the SVM classifier: Dataset 12.

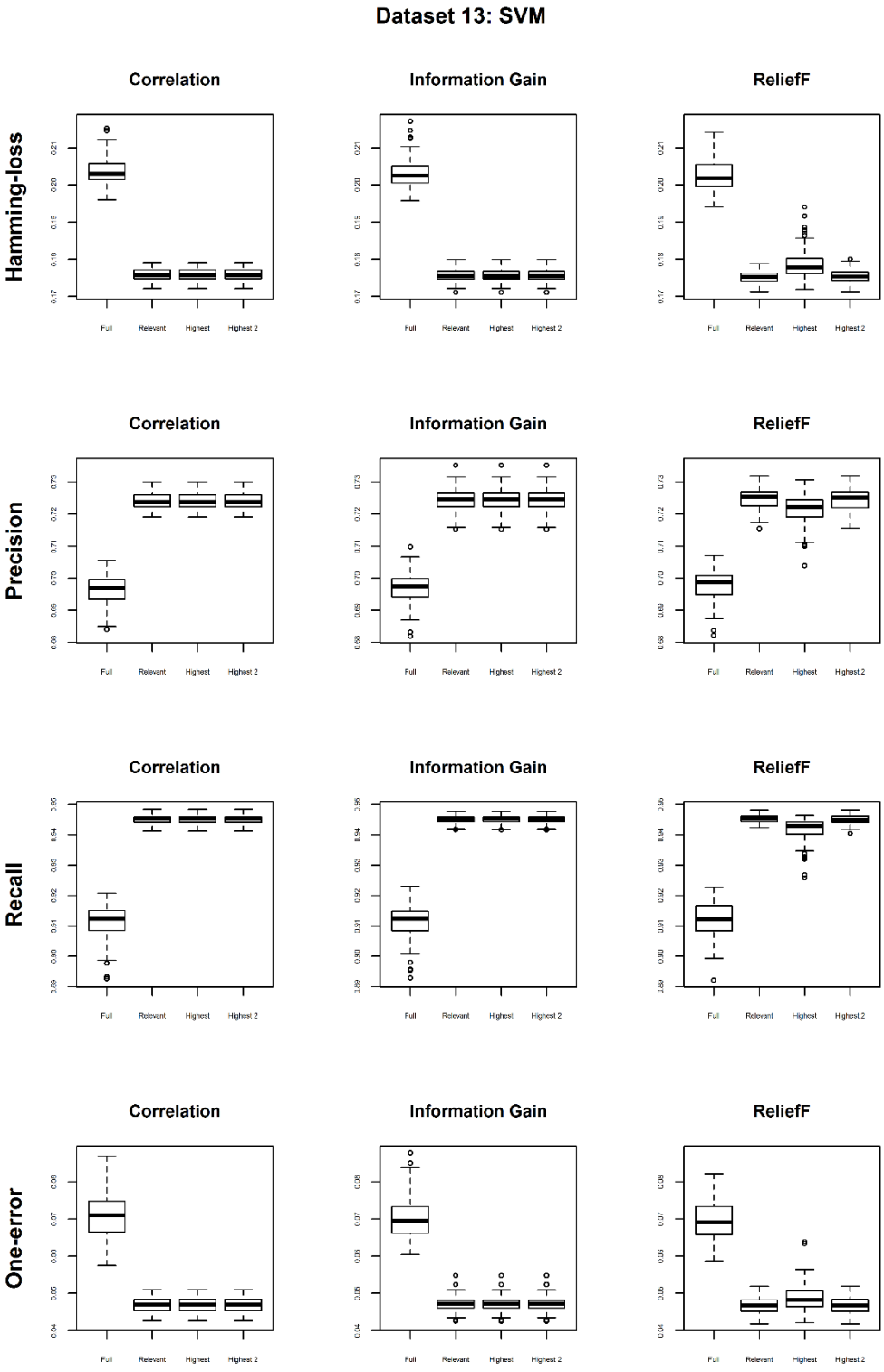


Figure D.13 Summary of results for the SVM classifier: Dataset 13.

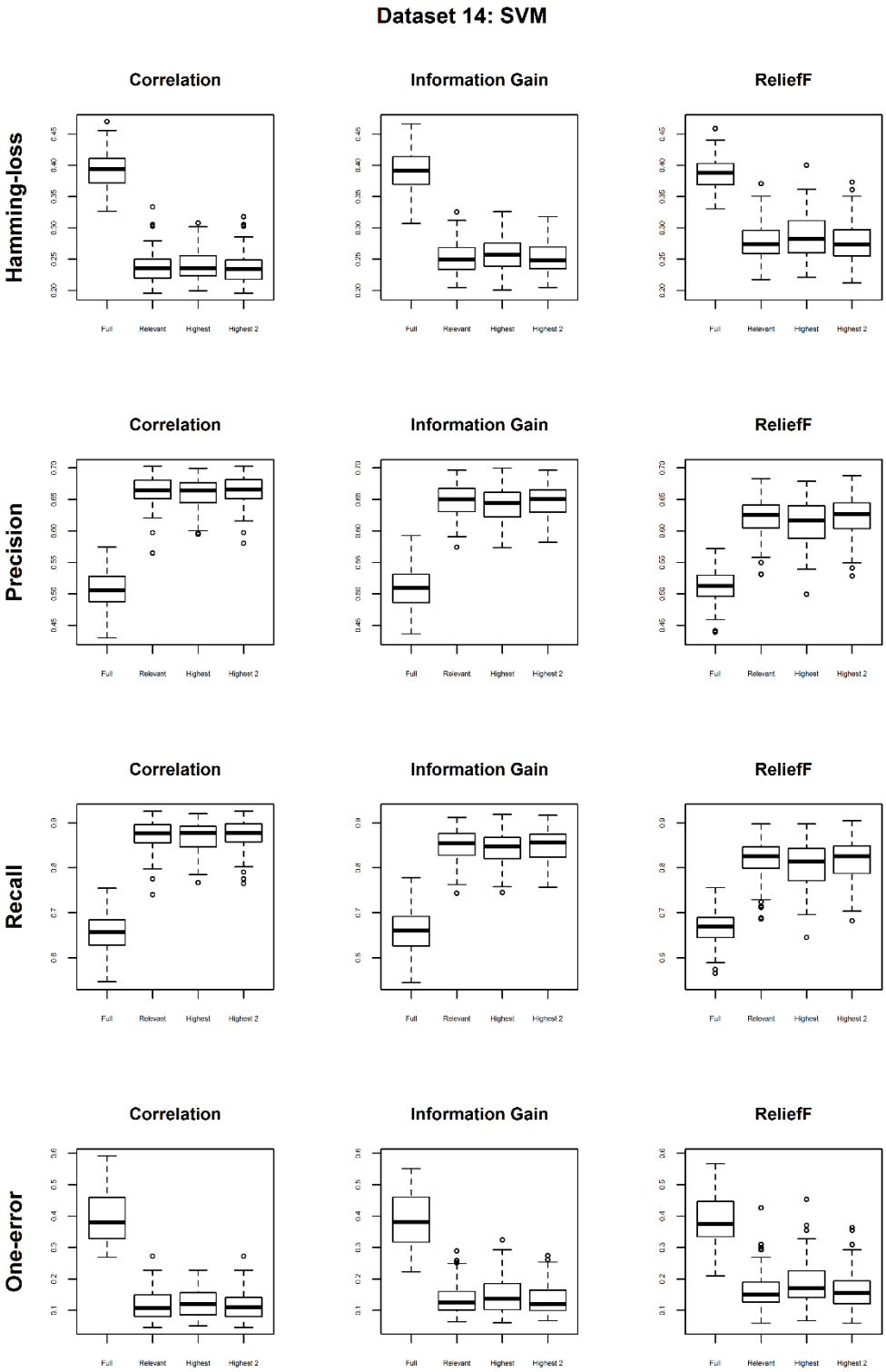


Figure D.14 Summary of results for the SVM classifier: Dataset 14.

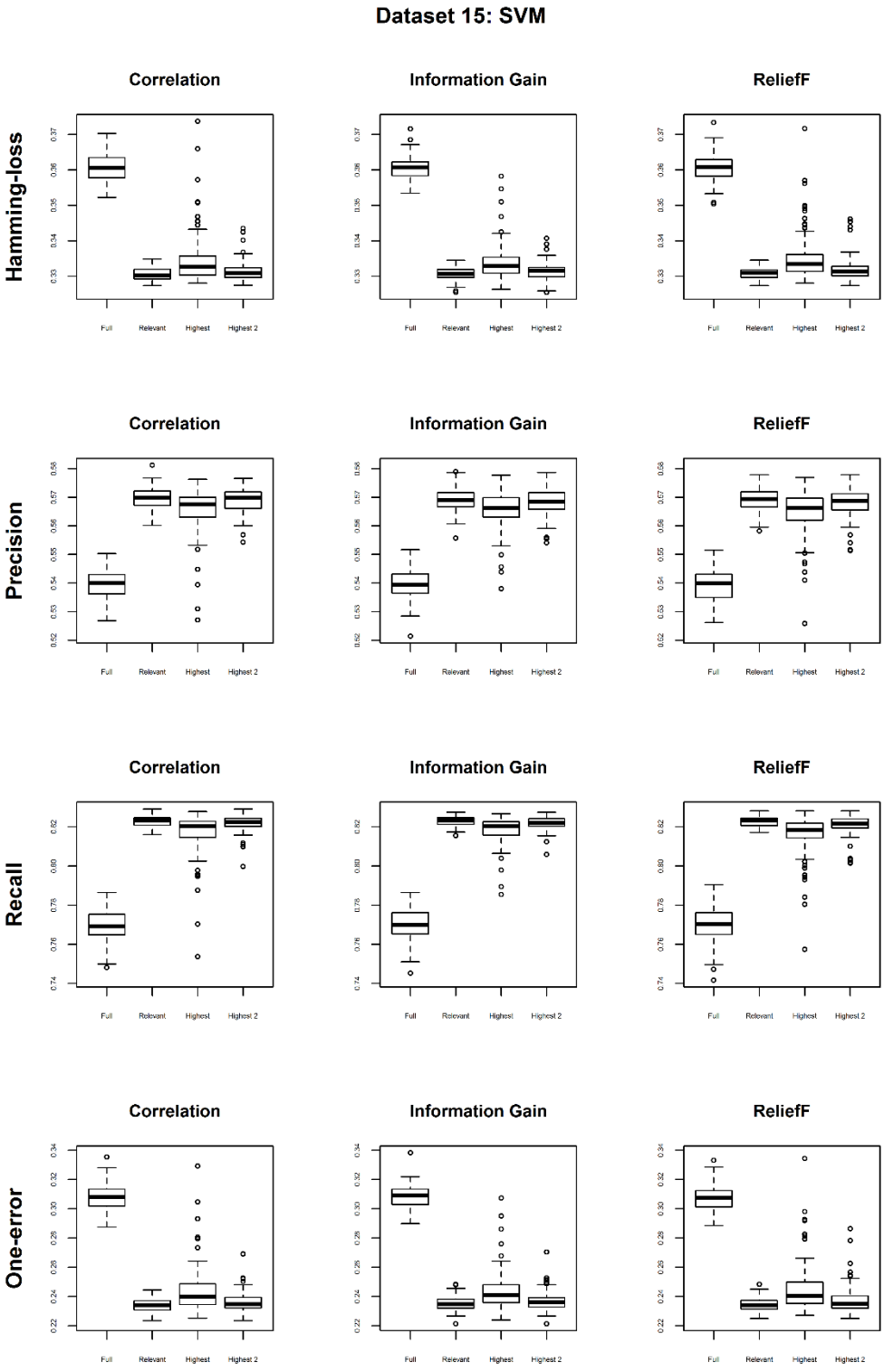


Figure D.15 Summary of results for the SVM classifier: Dataset 15.

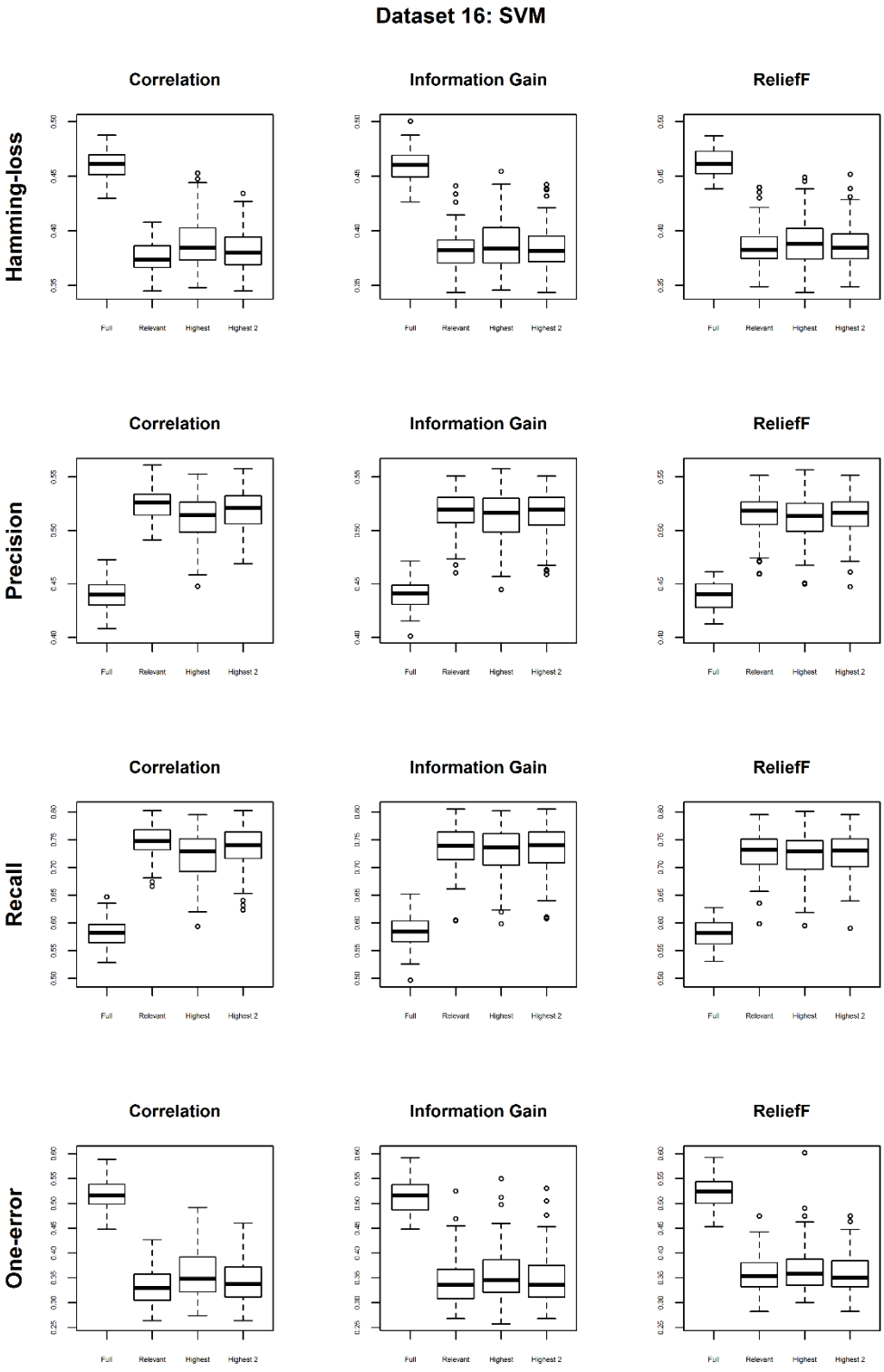


Figure D.16 Summary of results for the SVM classifier: Dataset 16.

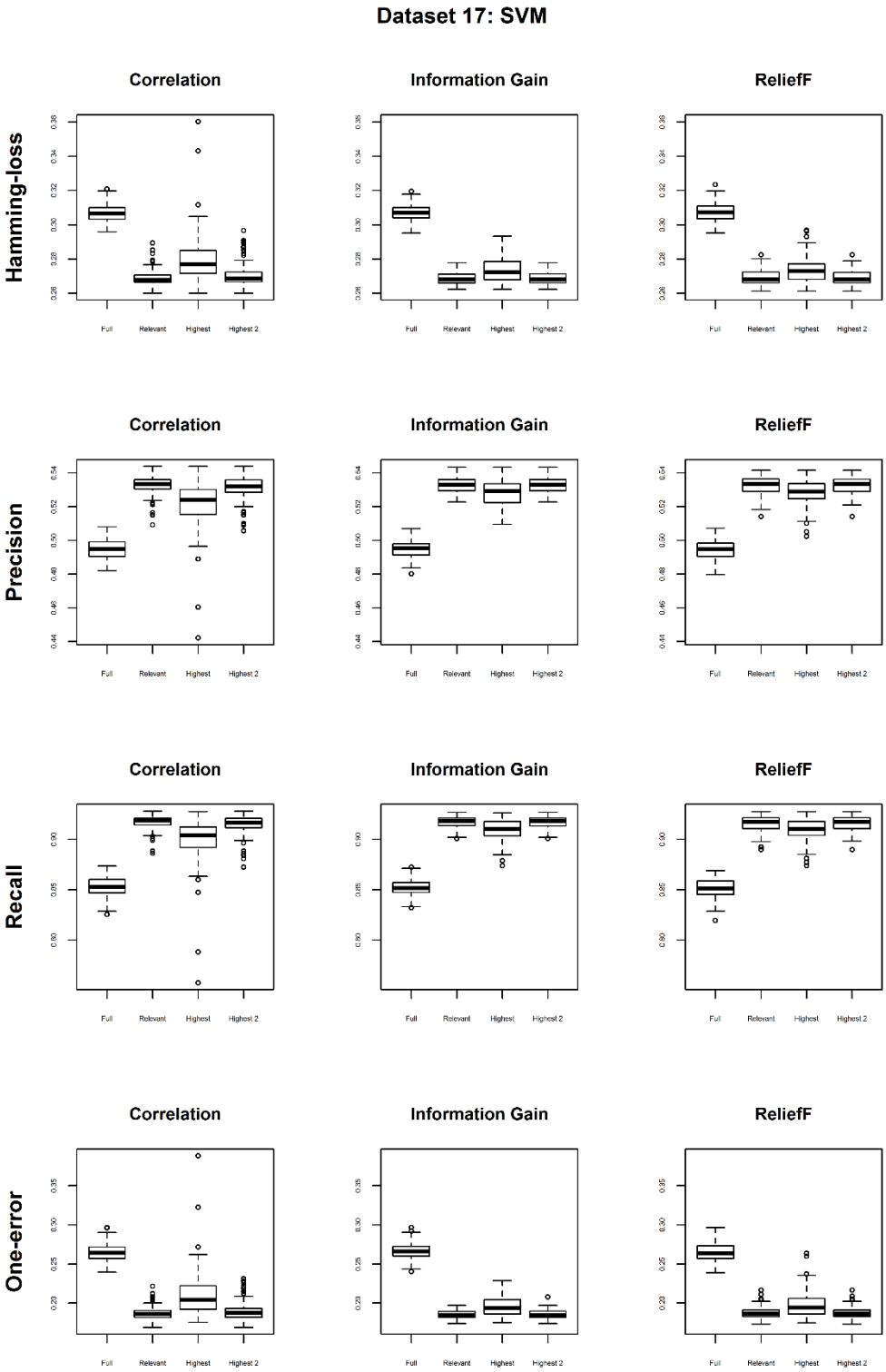


Figure D.17 Summary of results for the SVM classifier: Dataset 17.

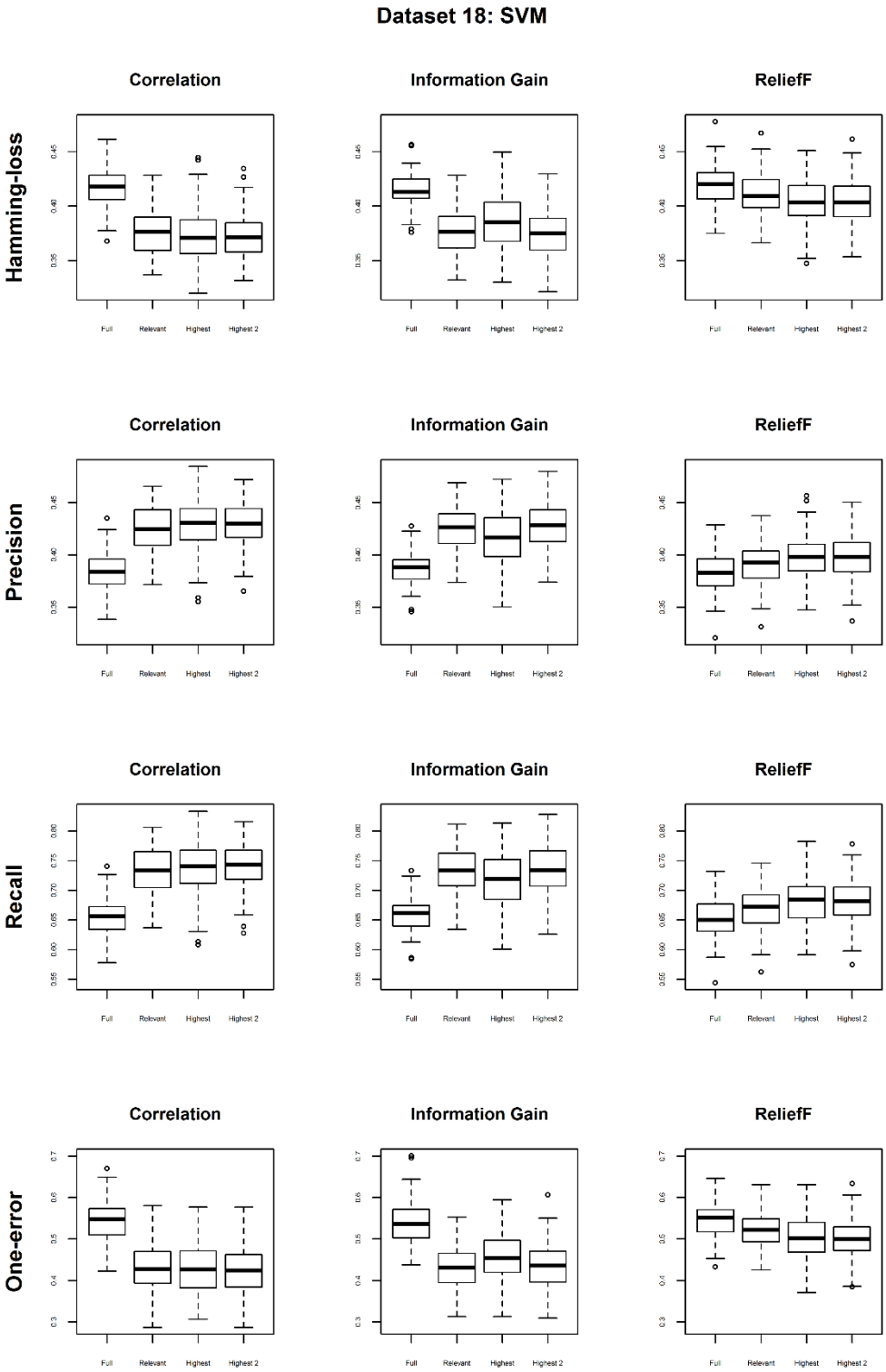


Figure D.18 Summary of results for the SVM classifier: Dataset 18.

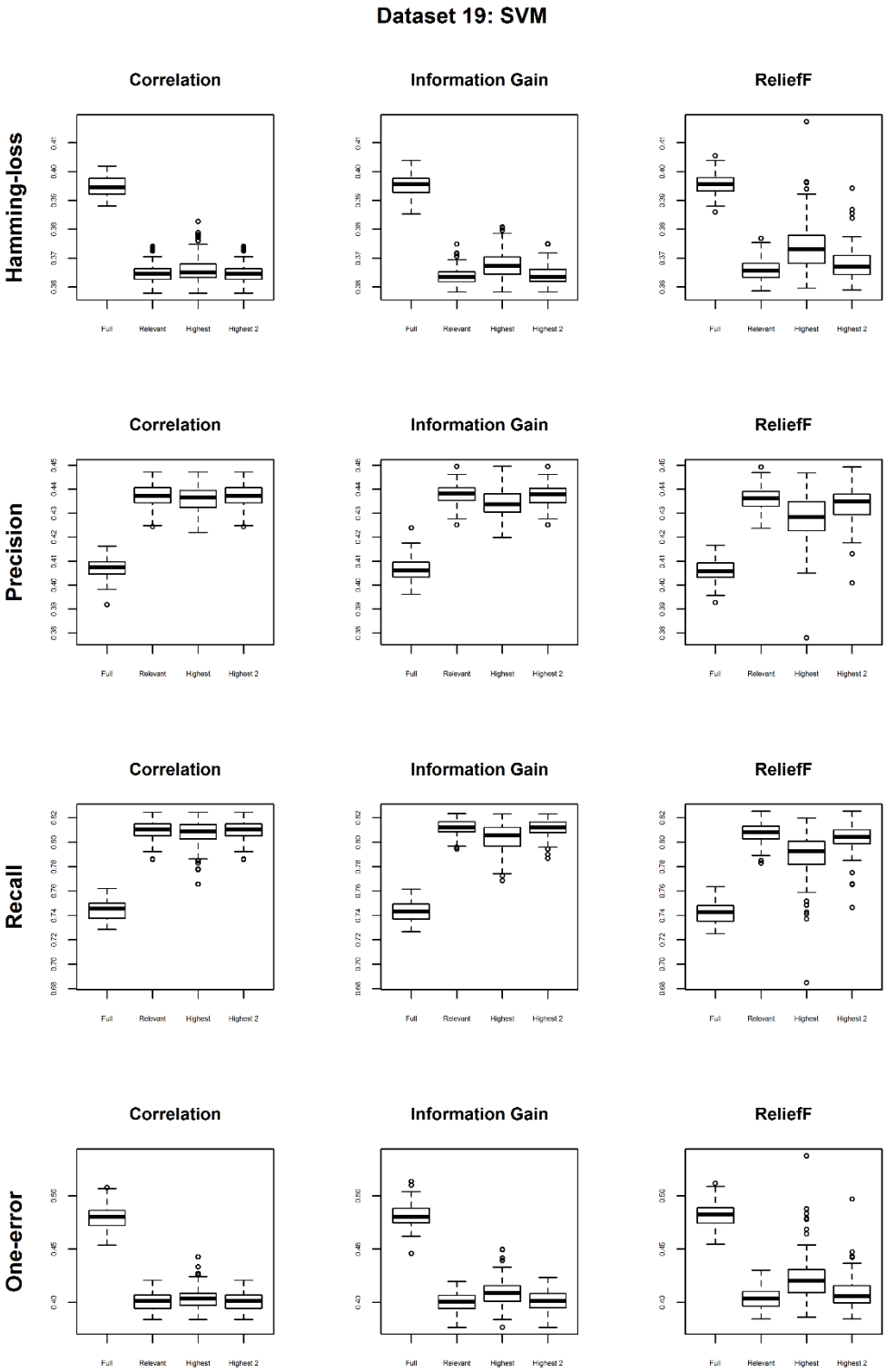


Figure D.19 Summary of results for the SVM classifier: Dataset 19.

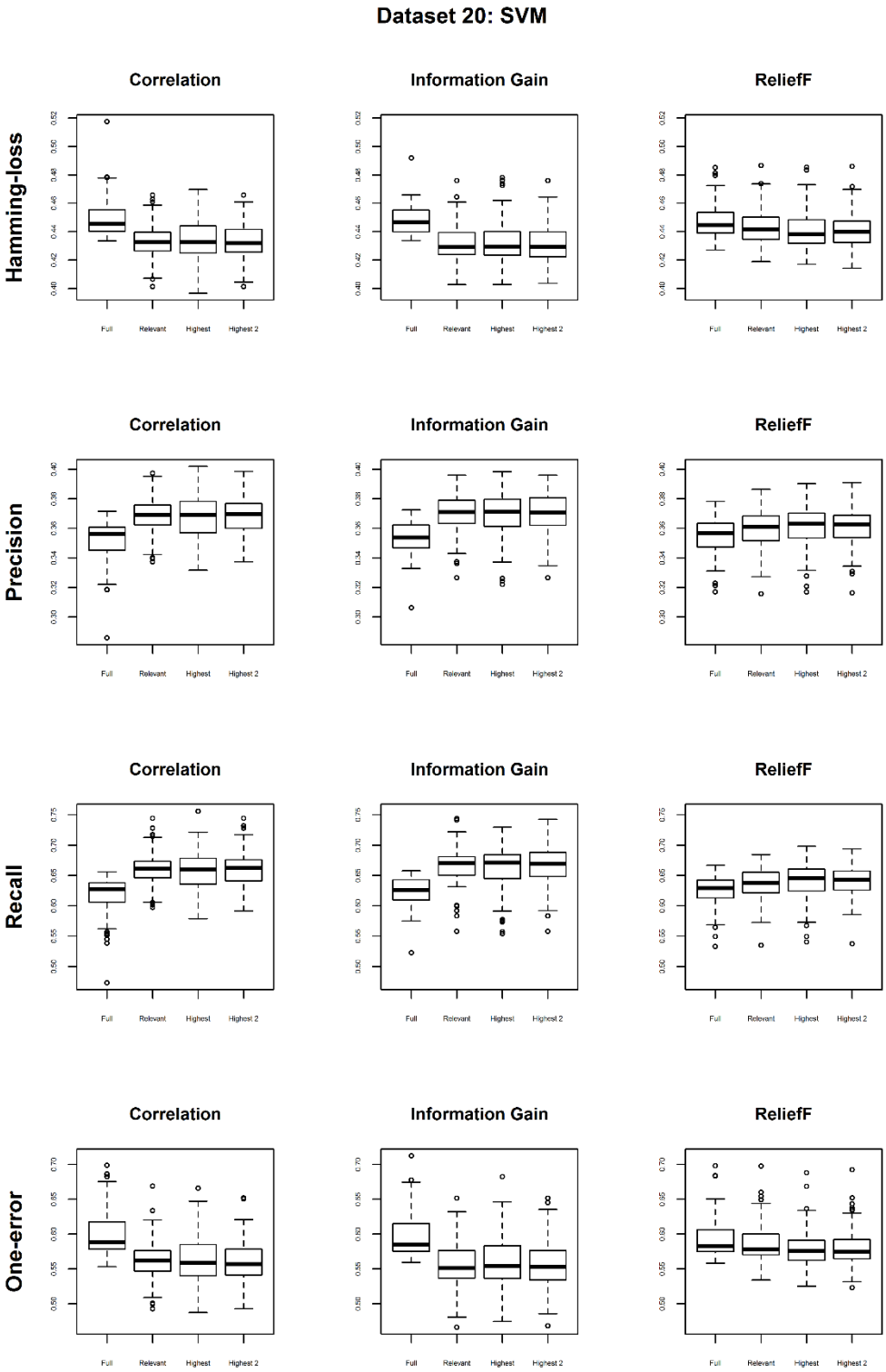


Figure D.20 Summary of results for the SVM classifier: Dataset 20.

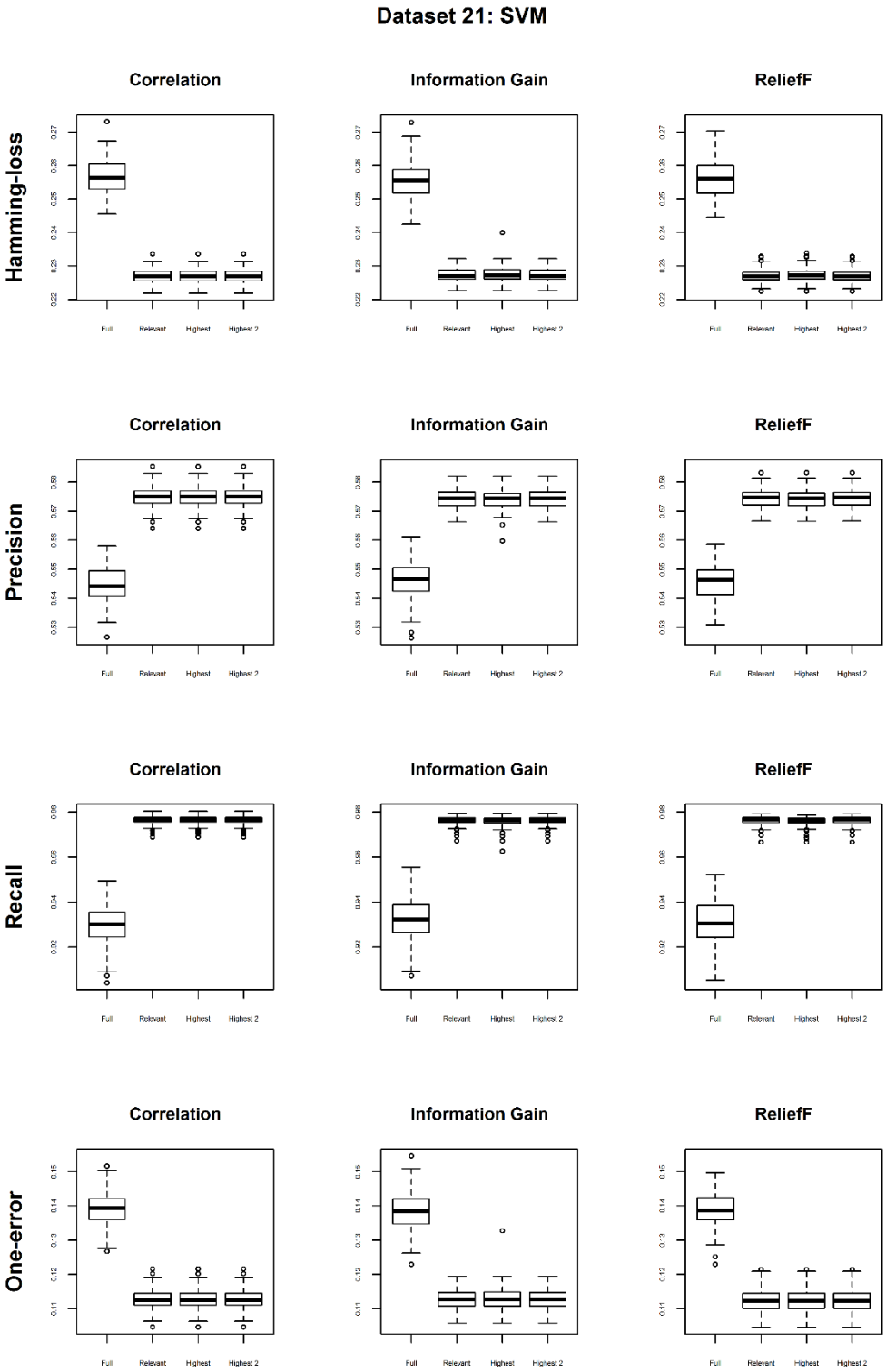


Figure D.21 Summary of results for the SVM classifier: Dataset 21.

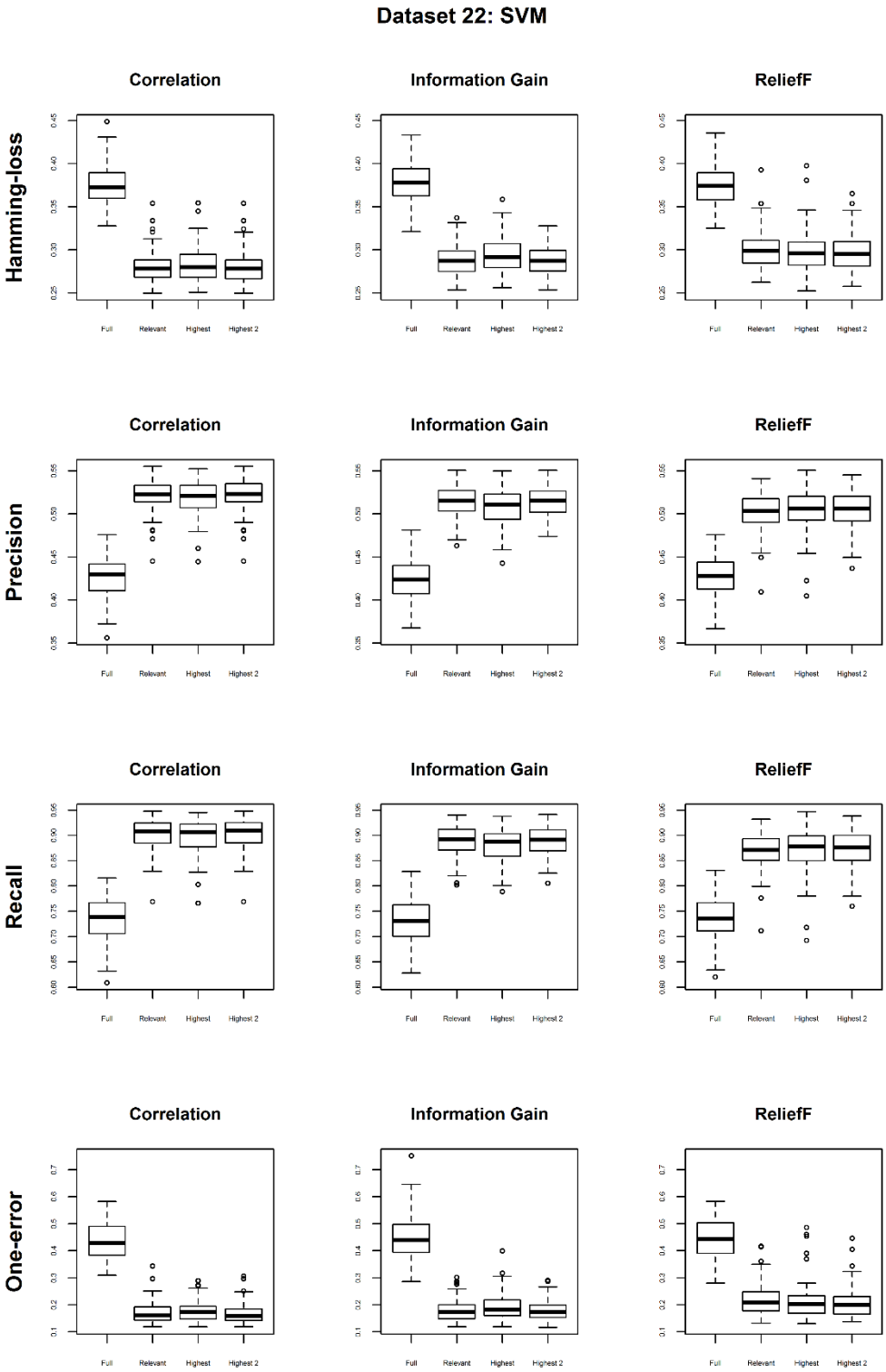


Figure D.22 Summary of results for the SVM classifier: Dataset 22.

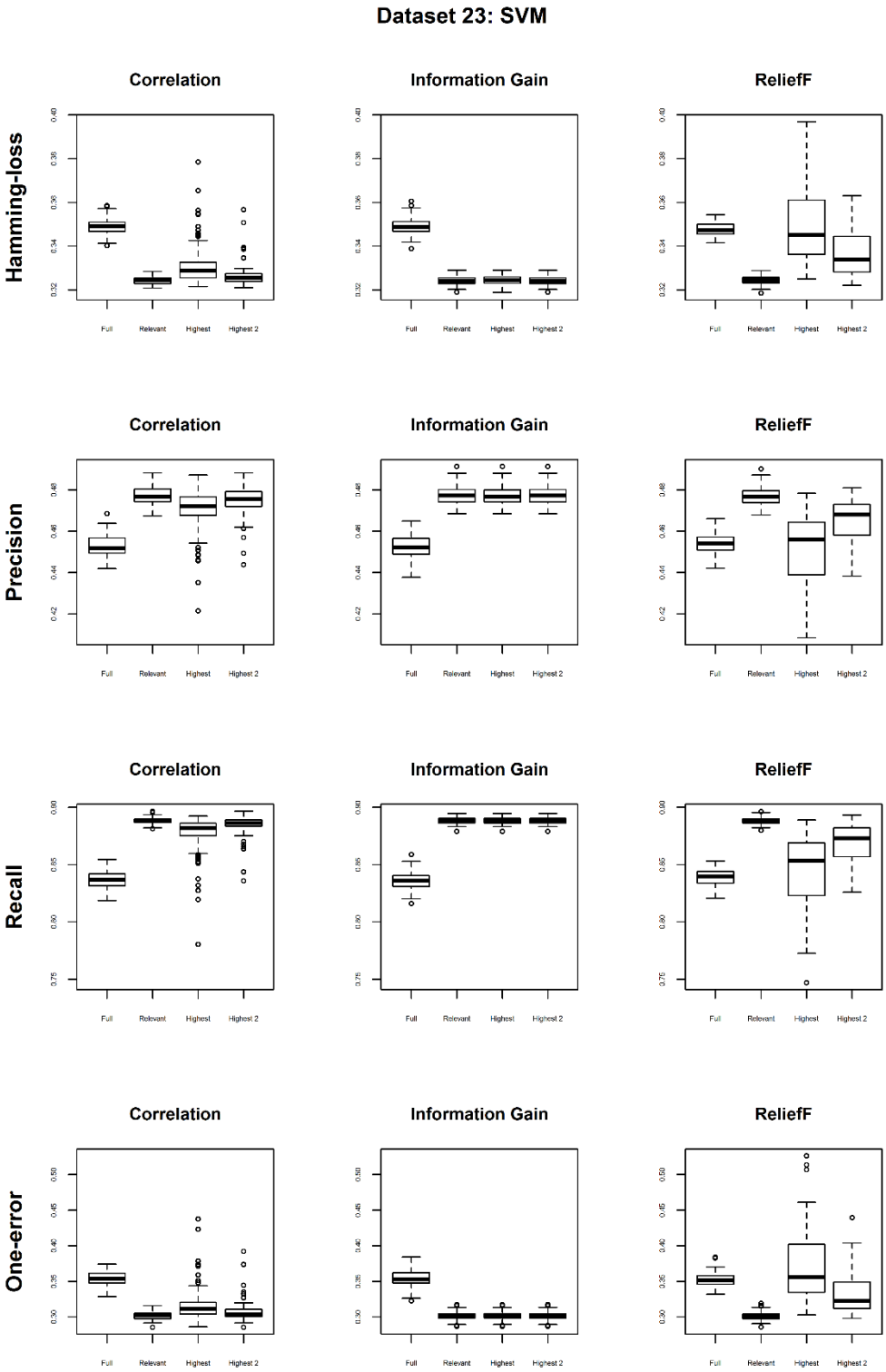


Figure D.23 Summary of results for the SVM classifier: Dataset 23.

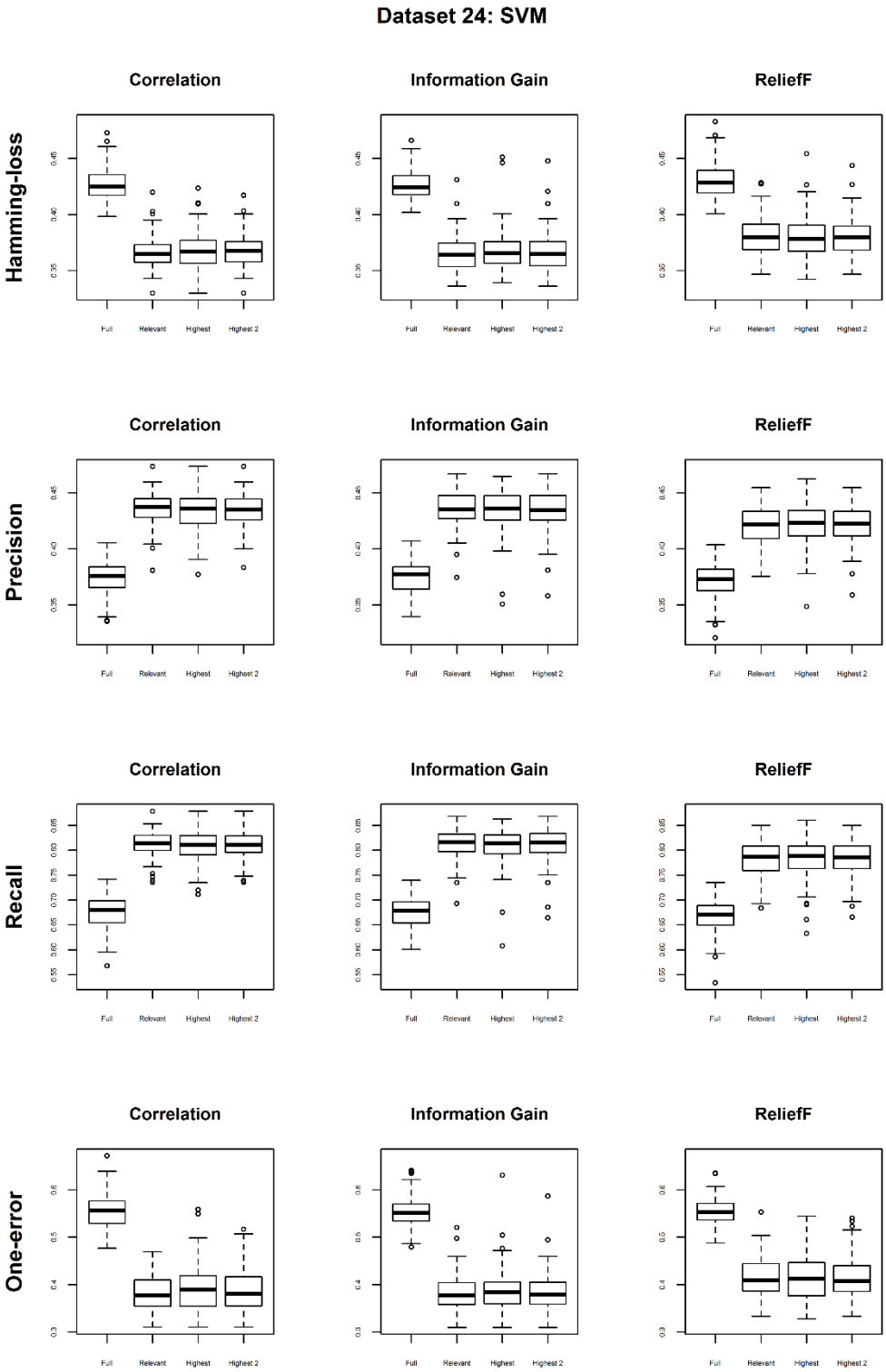


Figure D.24 Summary of results for the SVM classifier: Dataset 24.

APPENDIX E

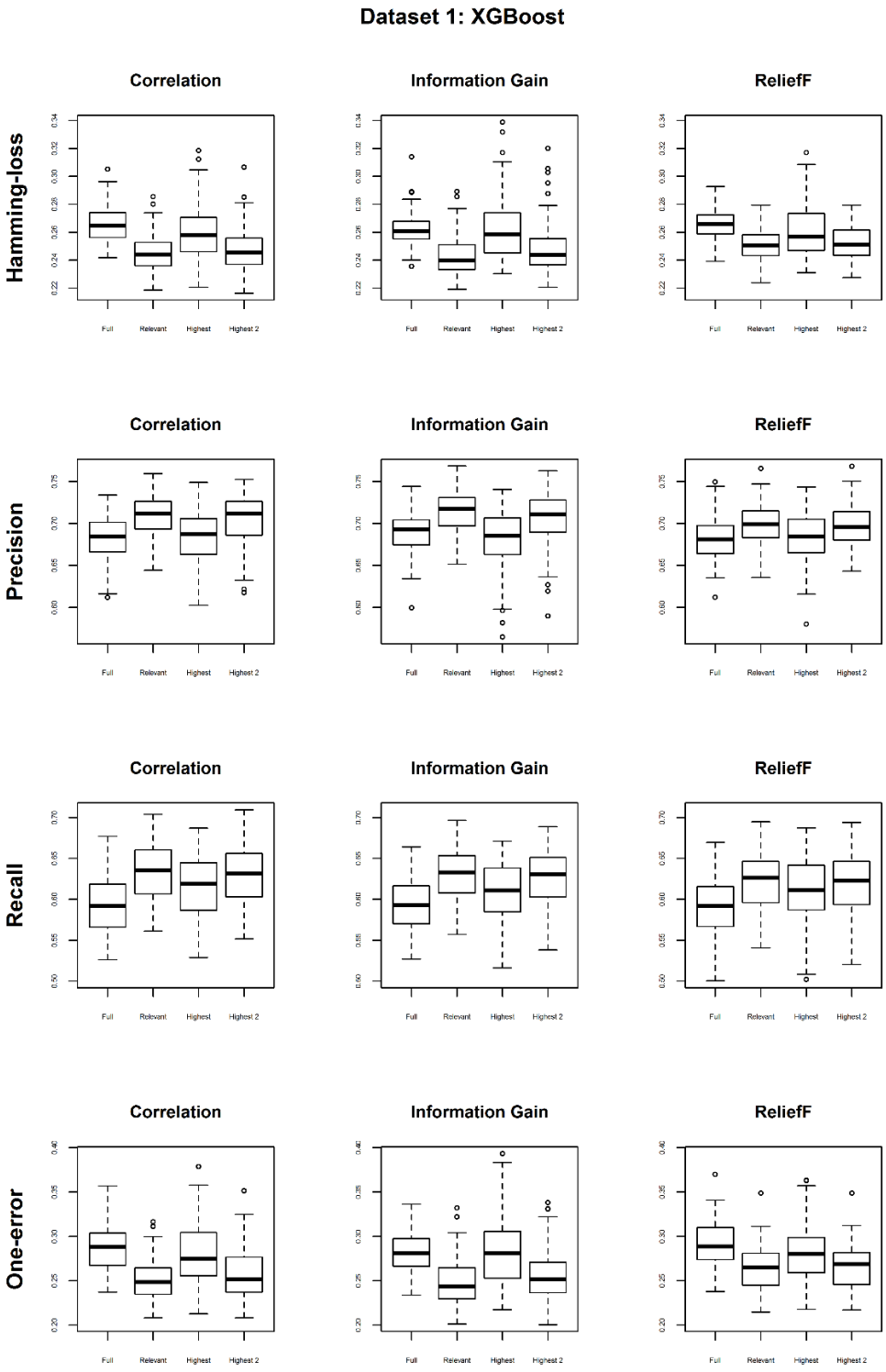


Figure E.1 Summary of results for the XGBoost classifier: Dataset 1.

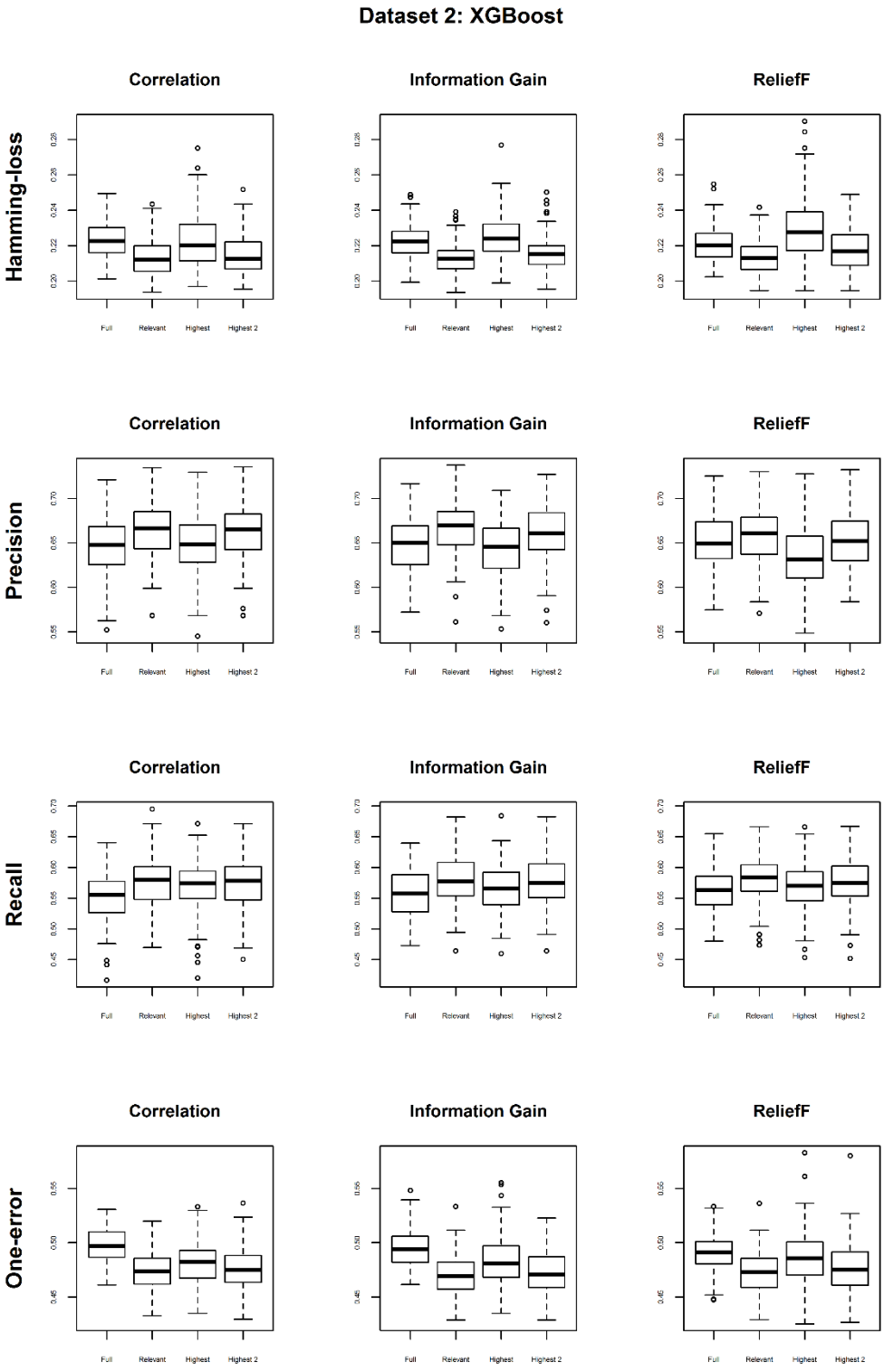


Figure E.2 Summary of results for the XGBoost classifier: Dataset 2.

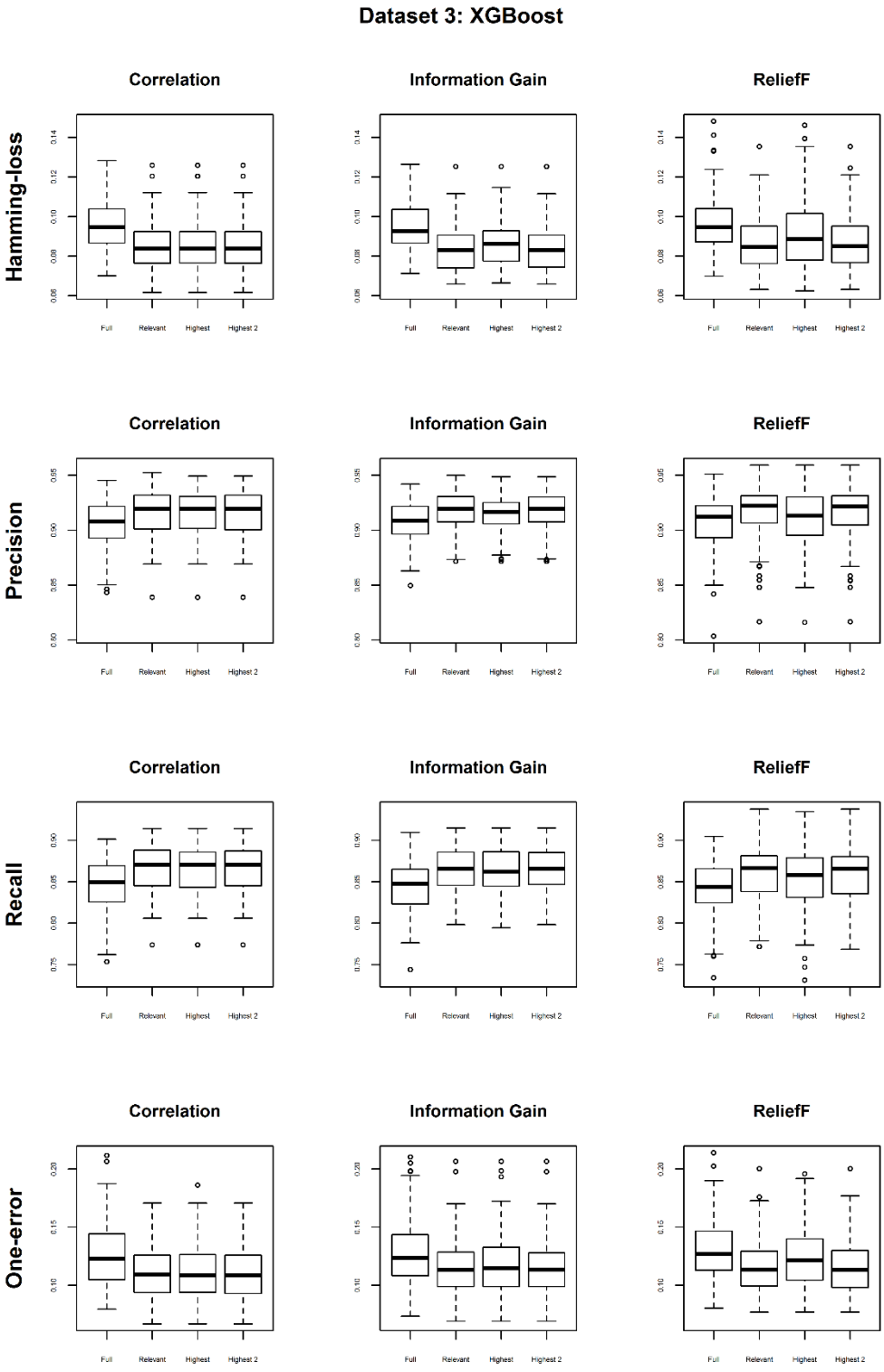


Figure E.3 Summary of results for the XGBoost classifier: Dataset 3.

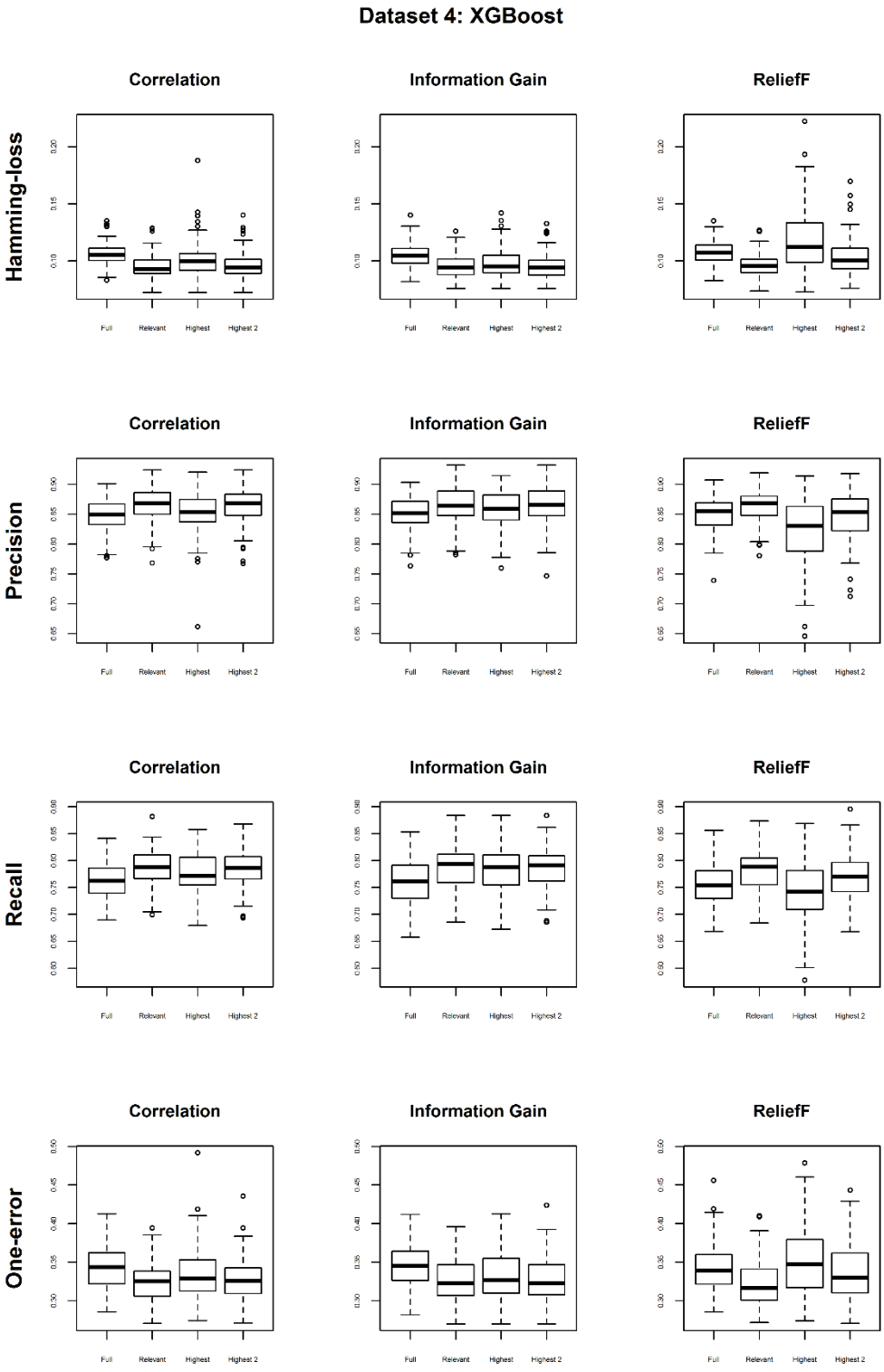


Figure E.4 Summary of results for the XGBoost classifier: Dataset 4.

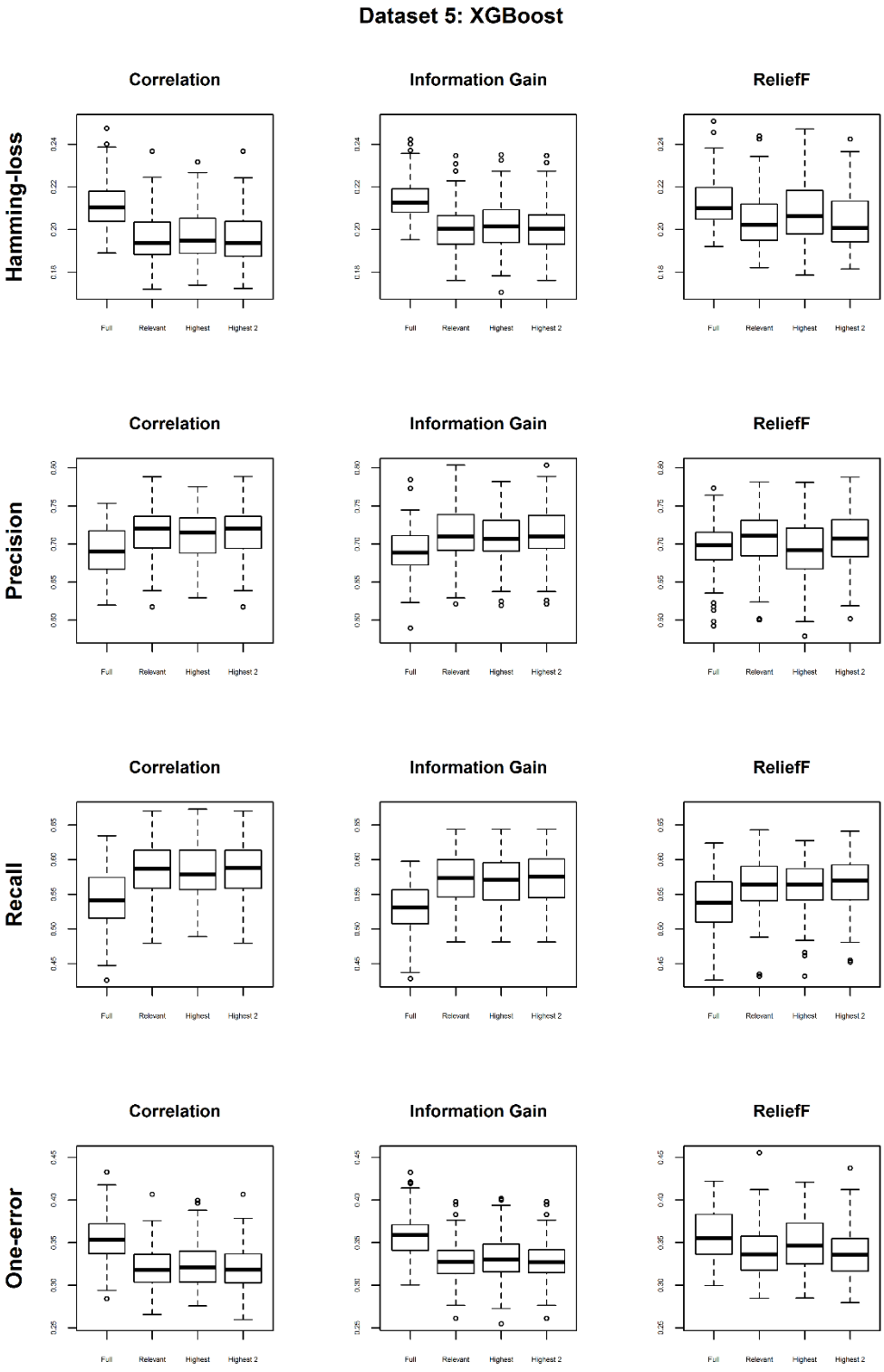


Figure E.5 Summary of results for the XGBoost classifier: Dataset 5.

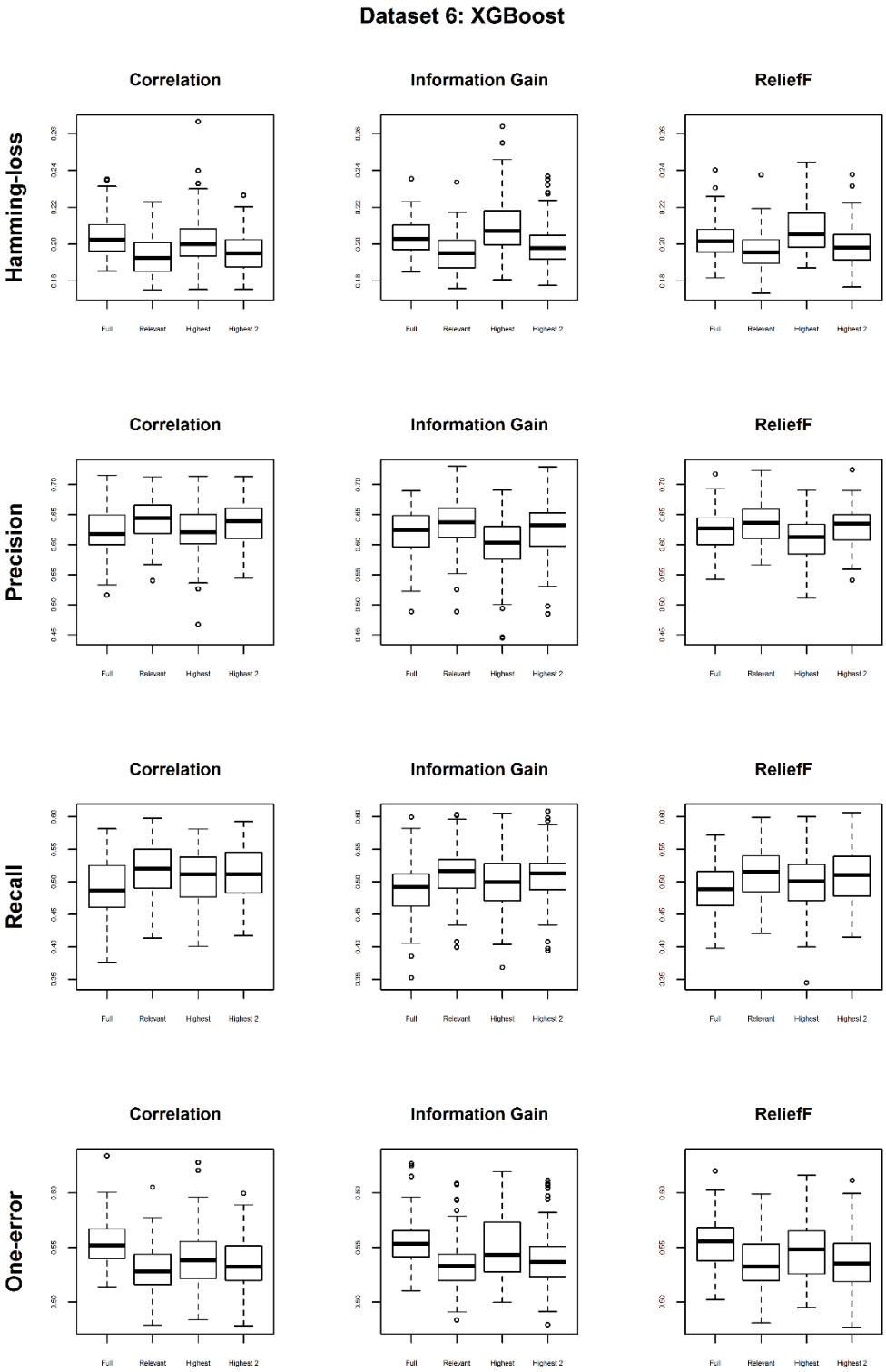


Figure E.6 Summary of results for the XGBoost classifier: Dataset 6.

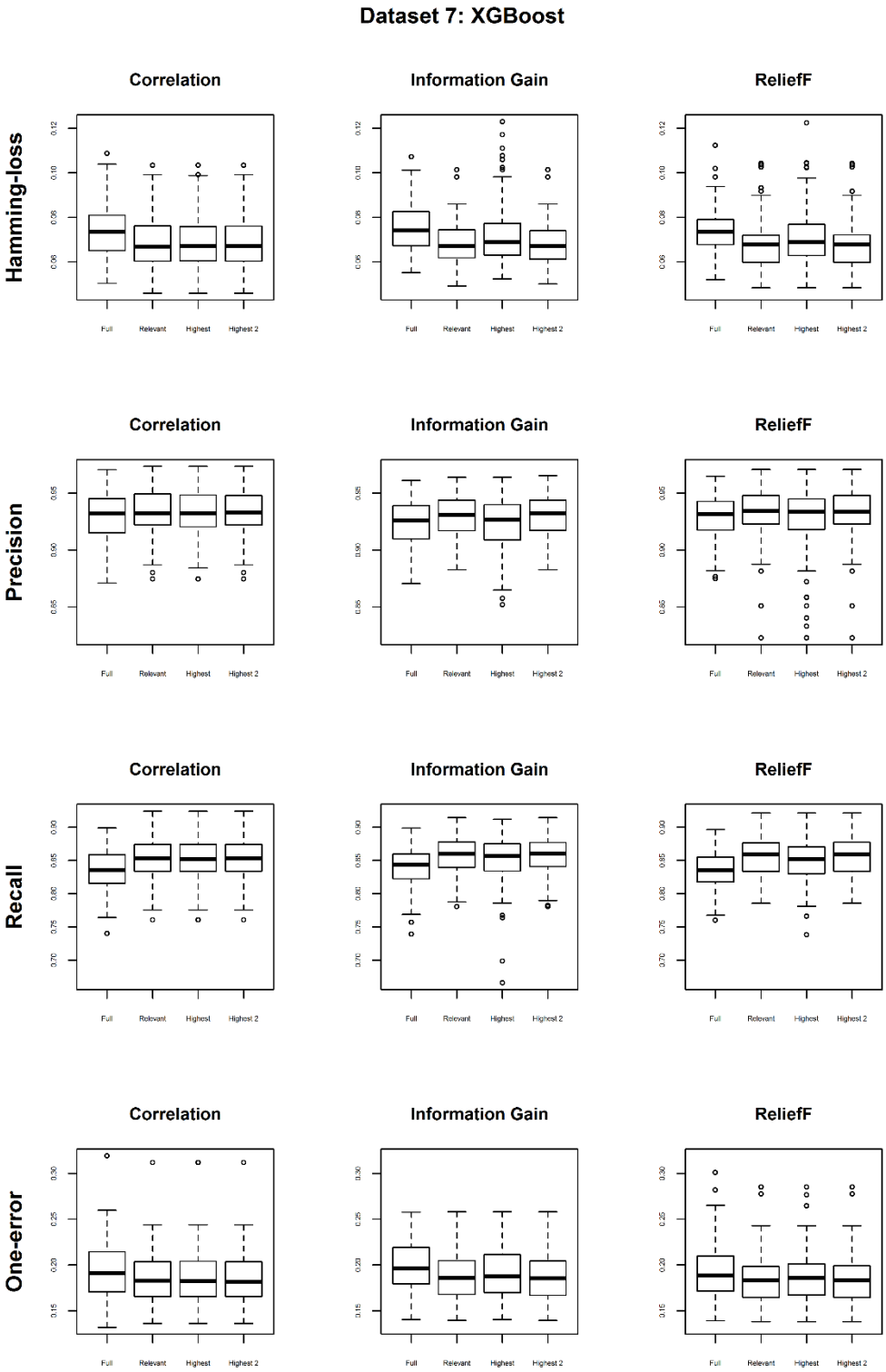


Figure E.7 Summary of results for the XGBoost classifier: Dataset 7.

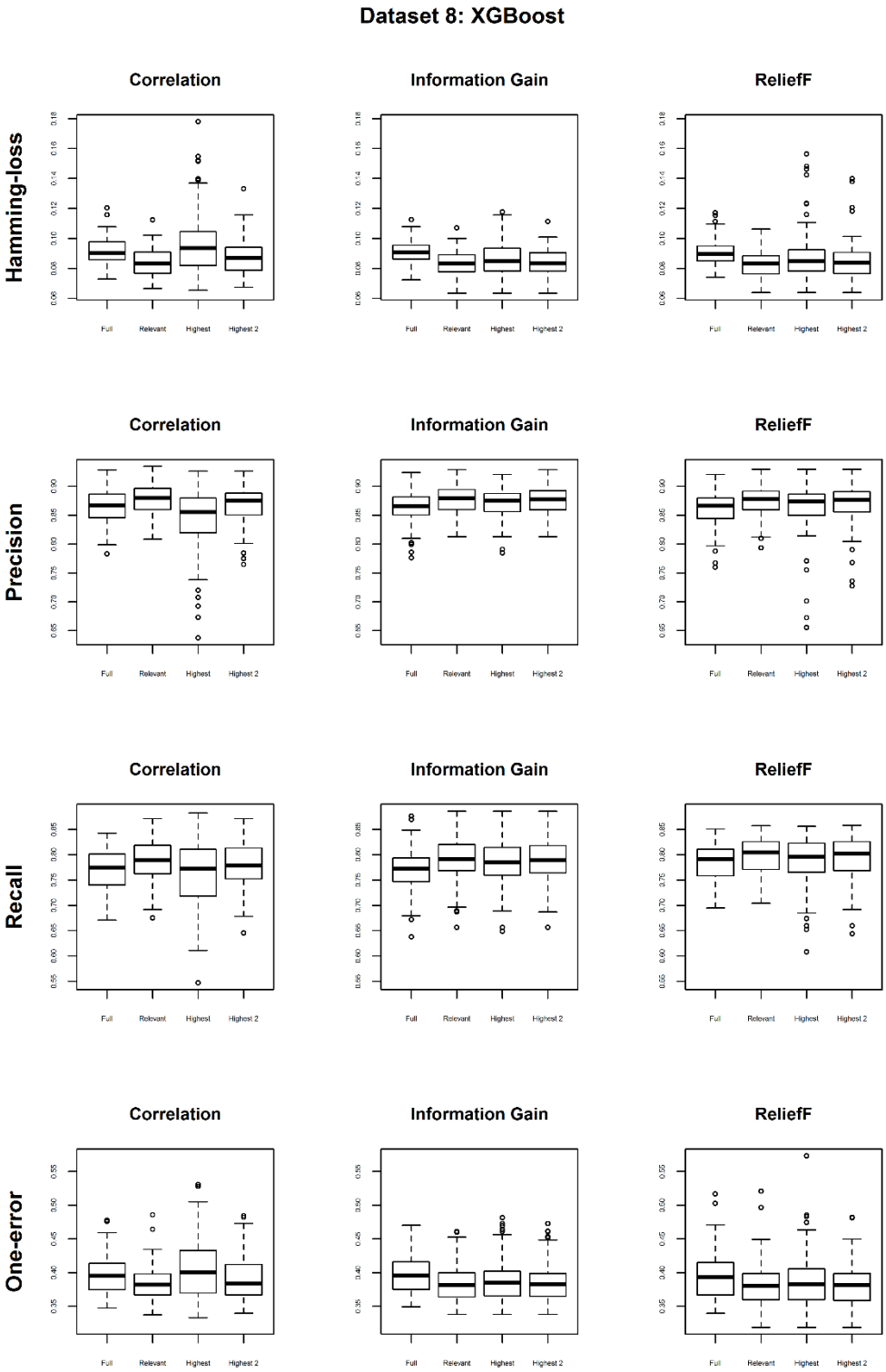


Figure E.8 Summary of results for the XGBoost classifier: Dataset 8.

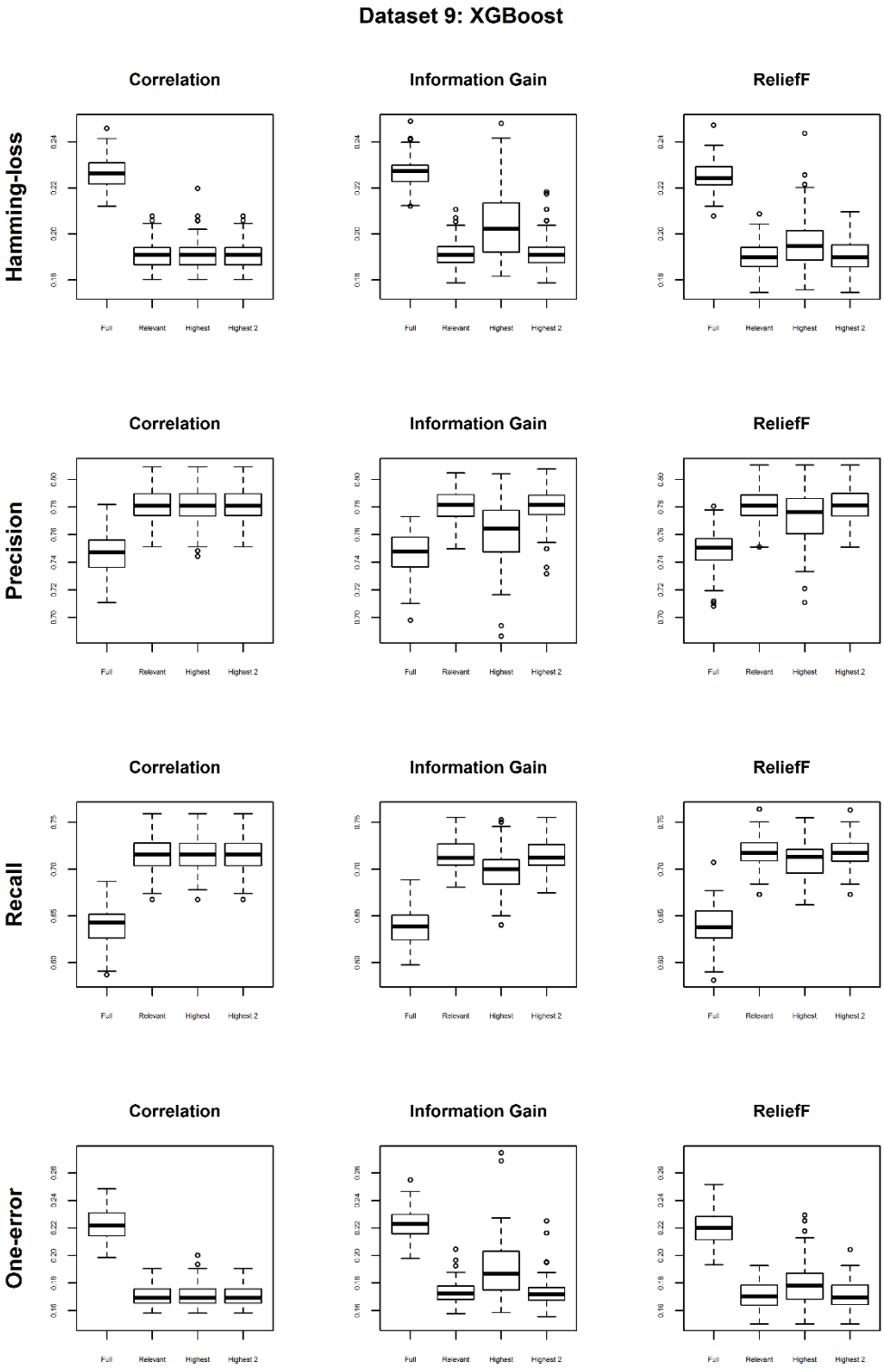


Figure E.9 Summary of results for the XGBoost classifier: Dataset 9.

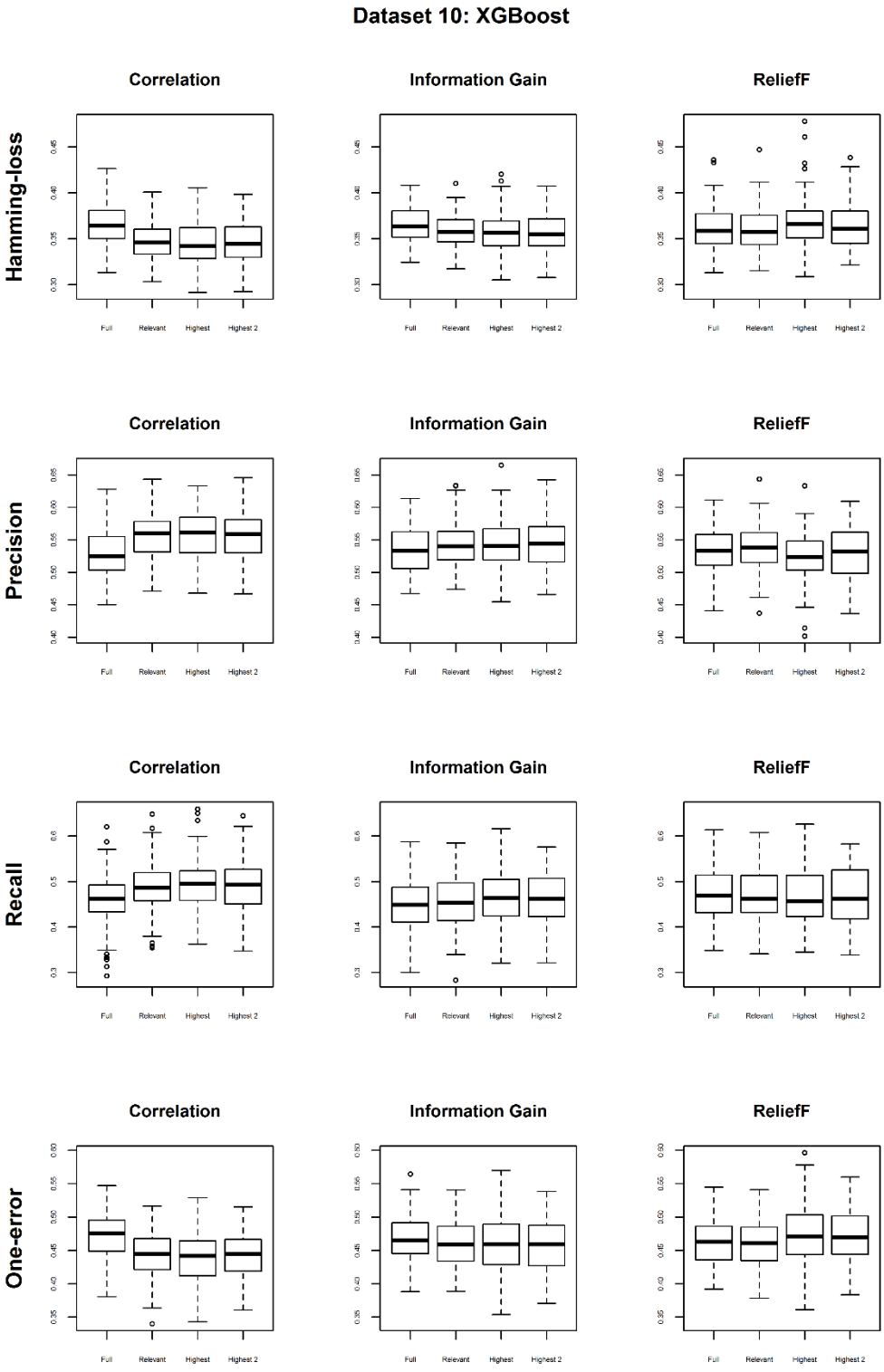


Figure E.10 Summary of results for the XGBoost classifier: Dataset 10.

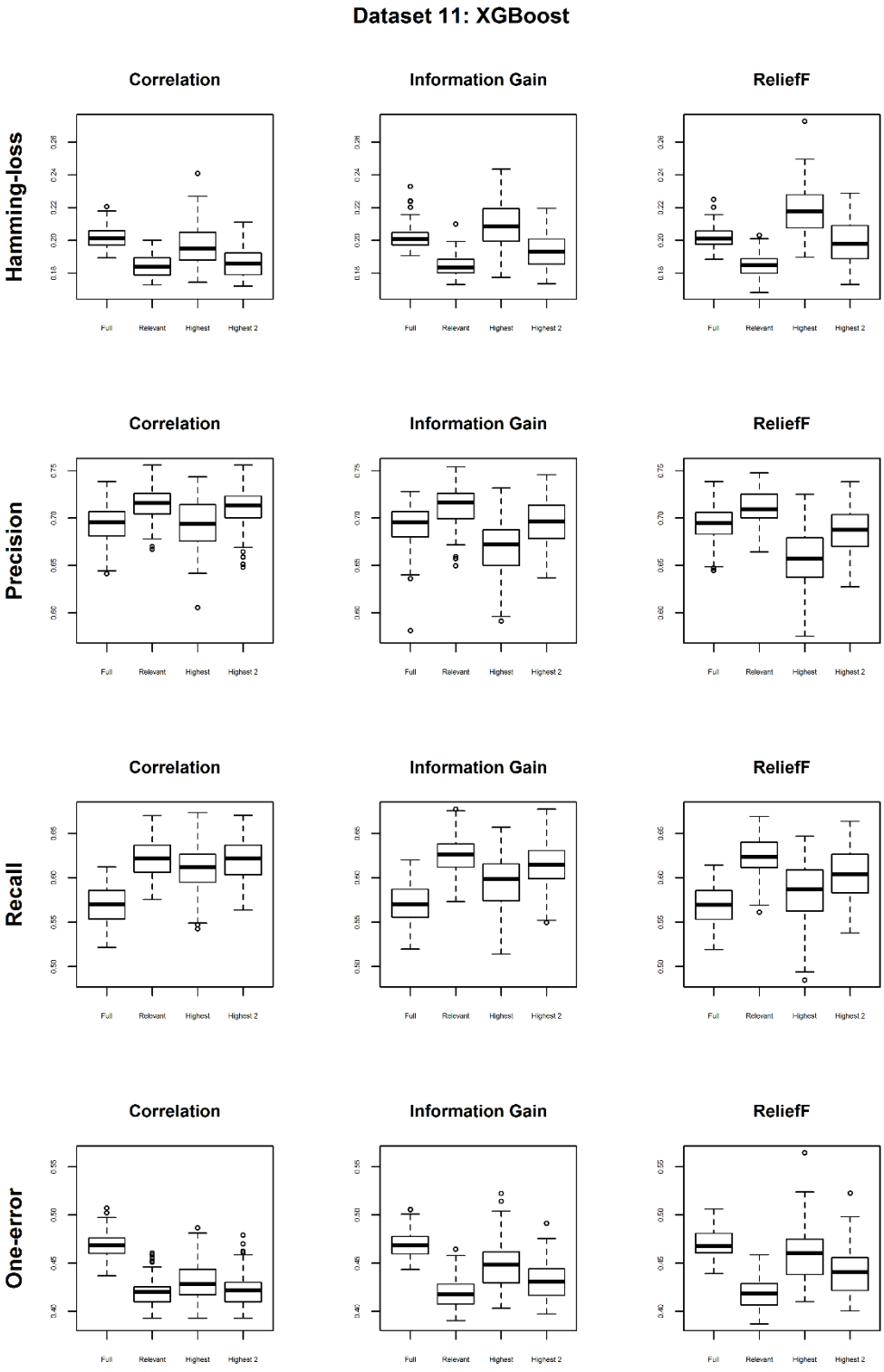


Figure E.11 Summary of results for the XGBoost classifier: Dataset 11.

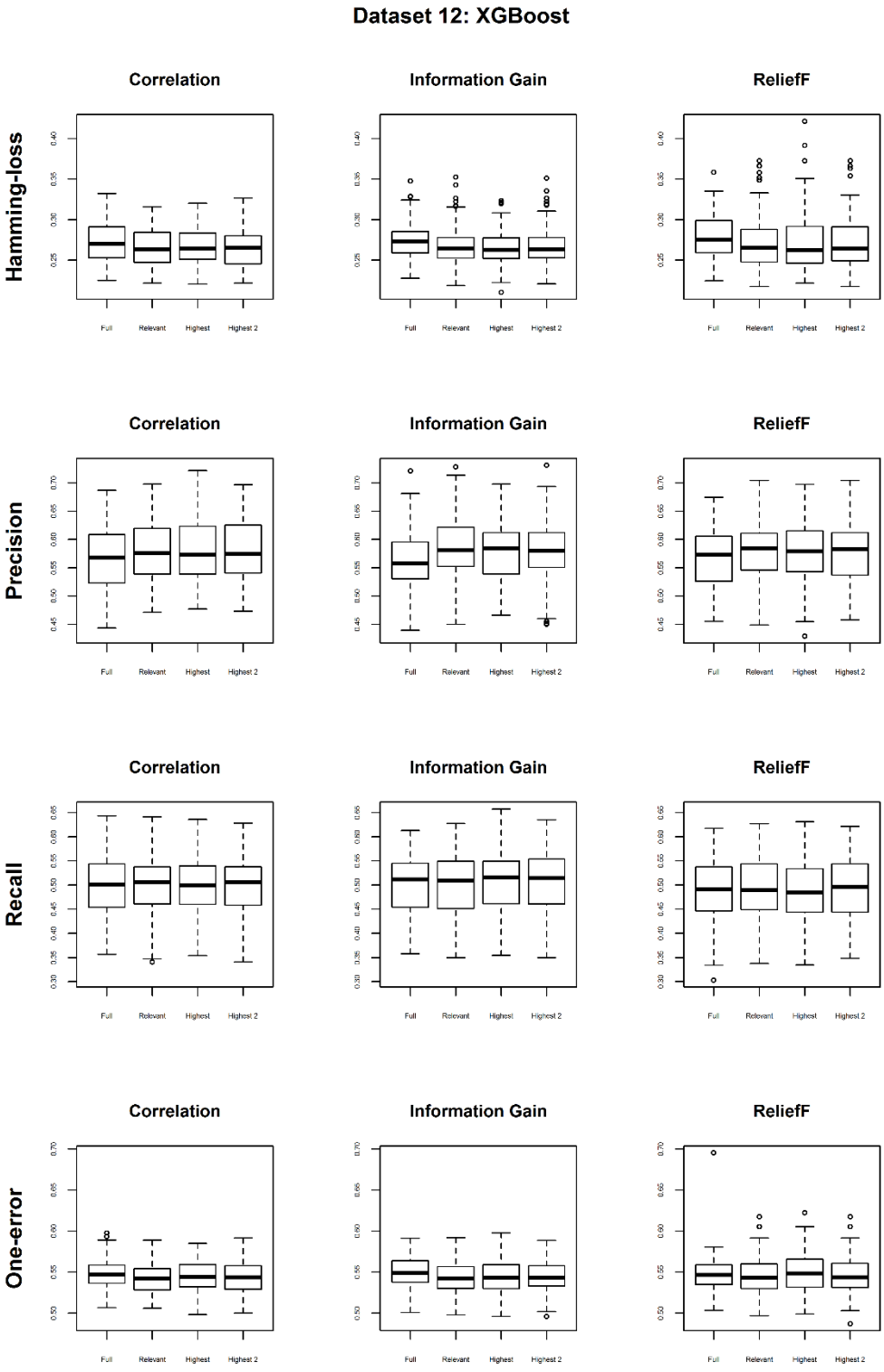


Figure E.12 Summary of results for the XGBoost classifier: Dataset 12.

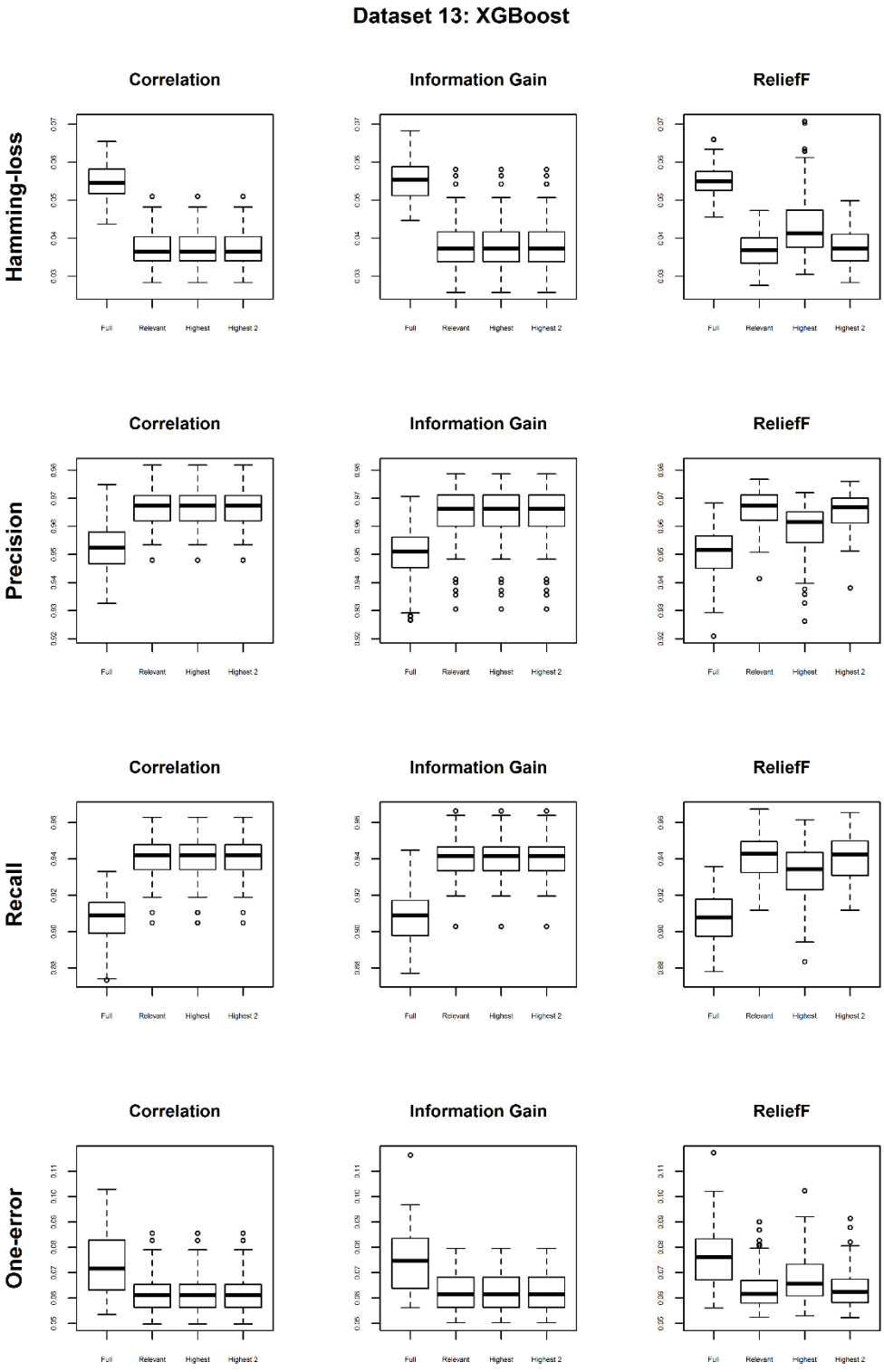


Figure E.13 Summary of results for the XGBoost classifier: Dataset 13.

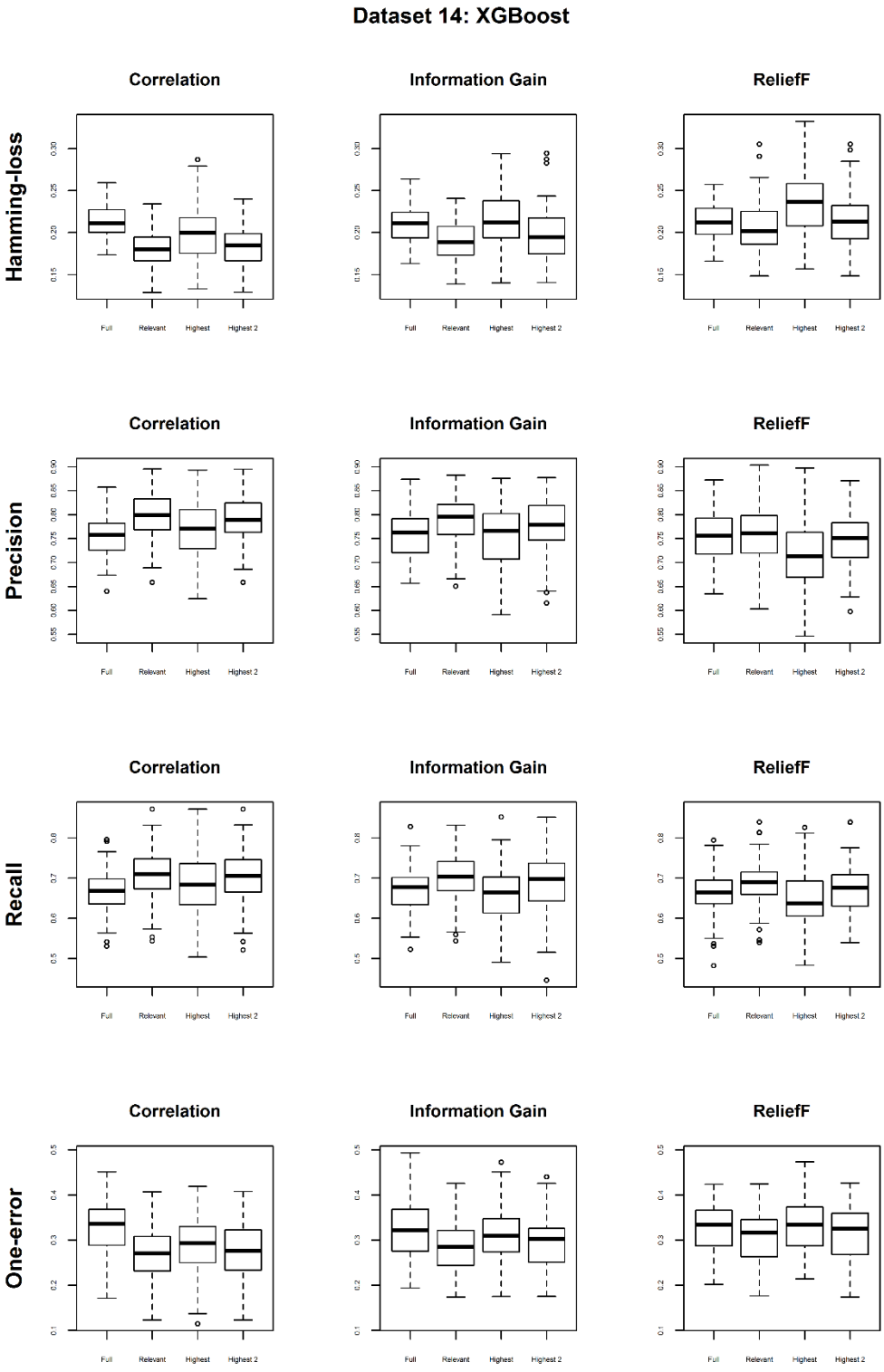


Figure E.14 Summary of results for the XGBoost classifier: Dataset 14.

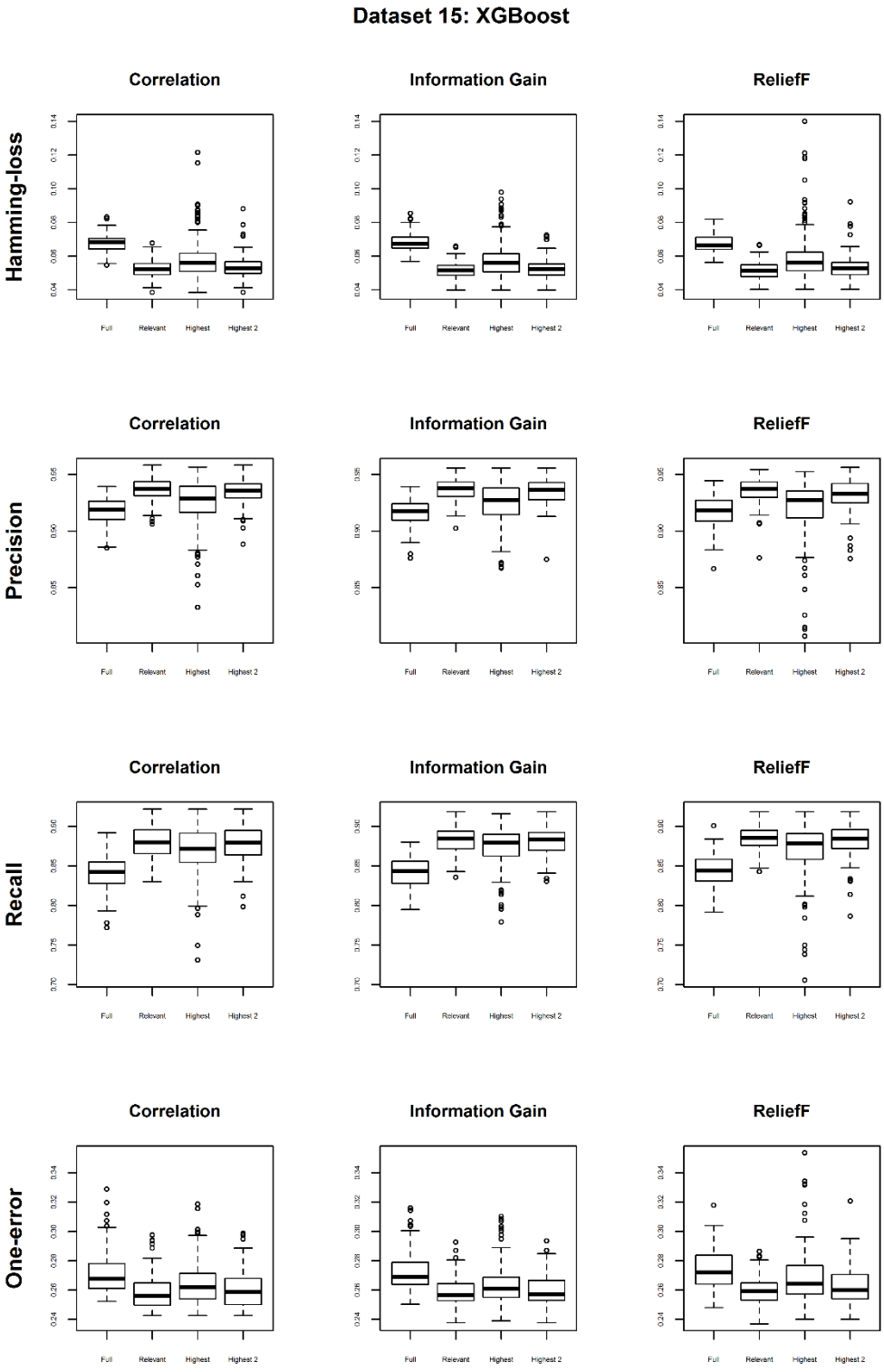


Figure E.15 Summary of results for the XGBoost classifier: Dataset 15.

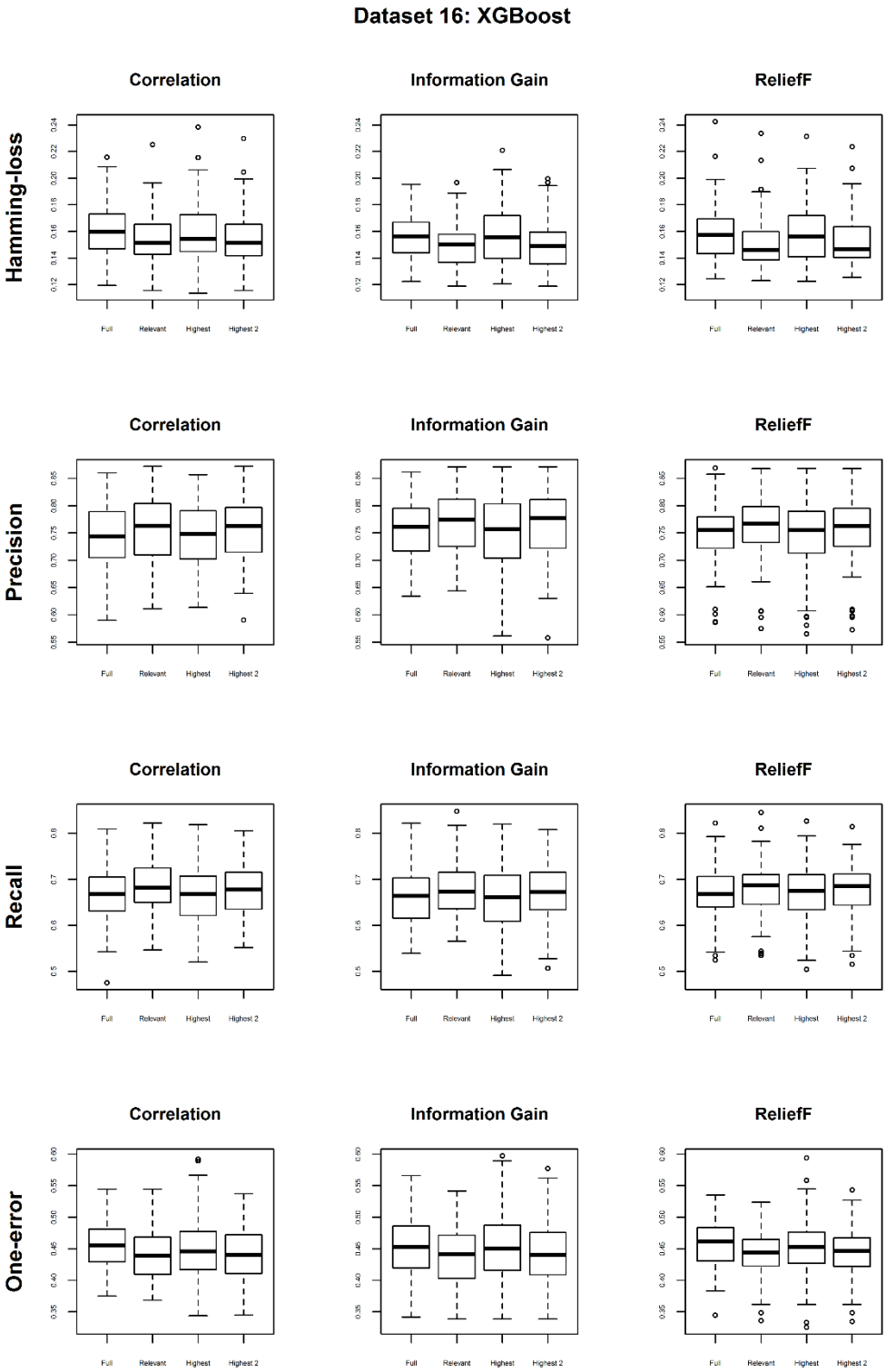


Figure E.16 Summary of results for the XGBoost classifier: Dataset 16.

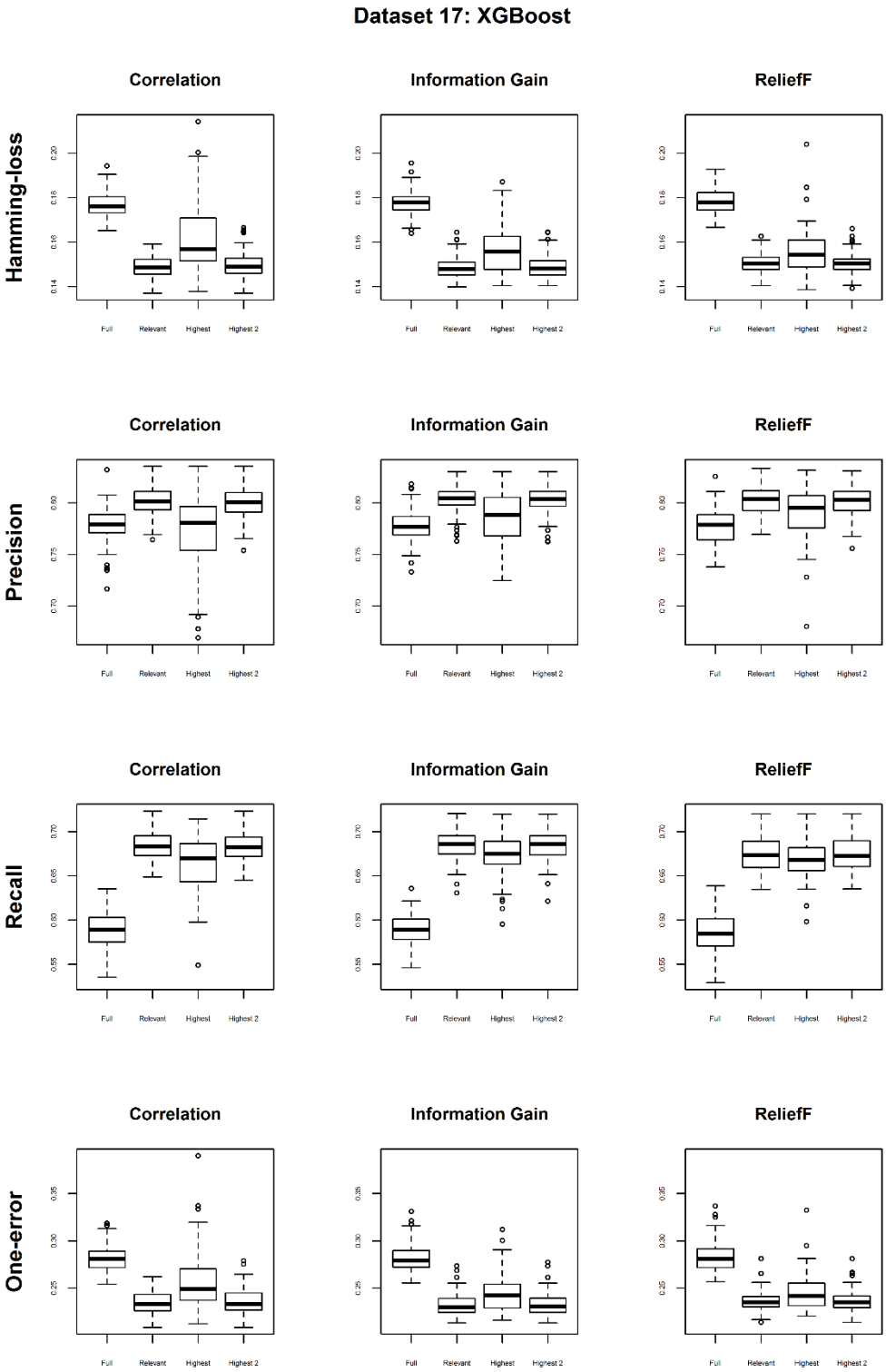


Figure E.17 Summary of results for the XGBoost classifier: Dataset 17.

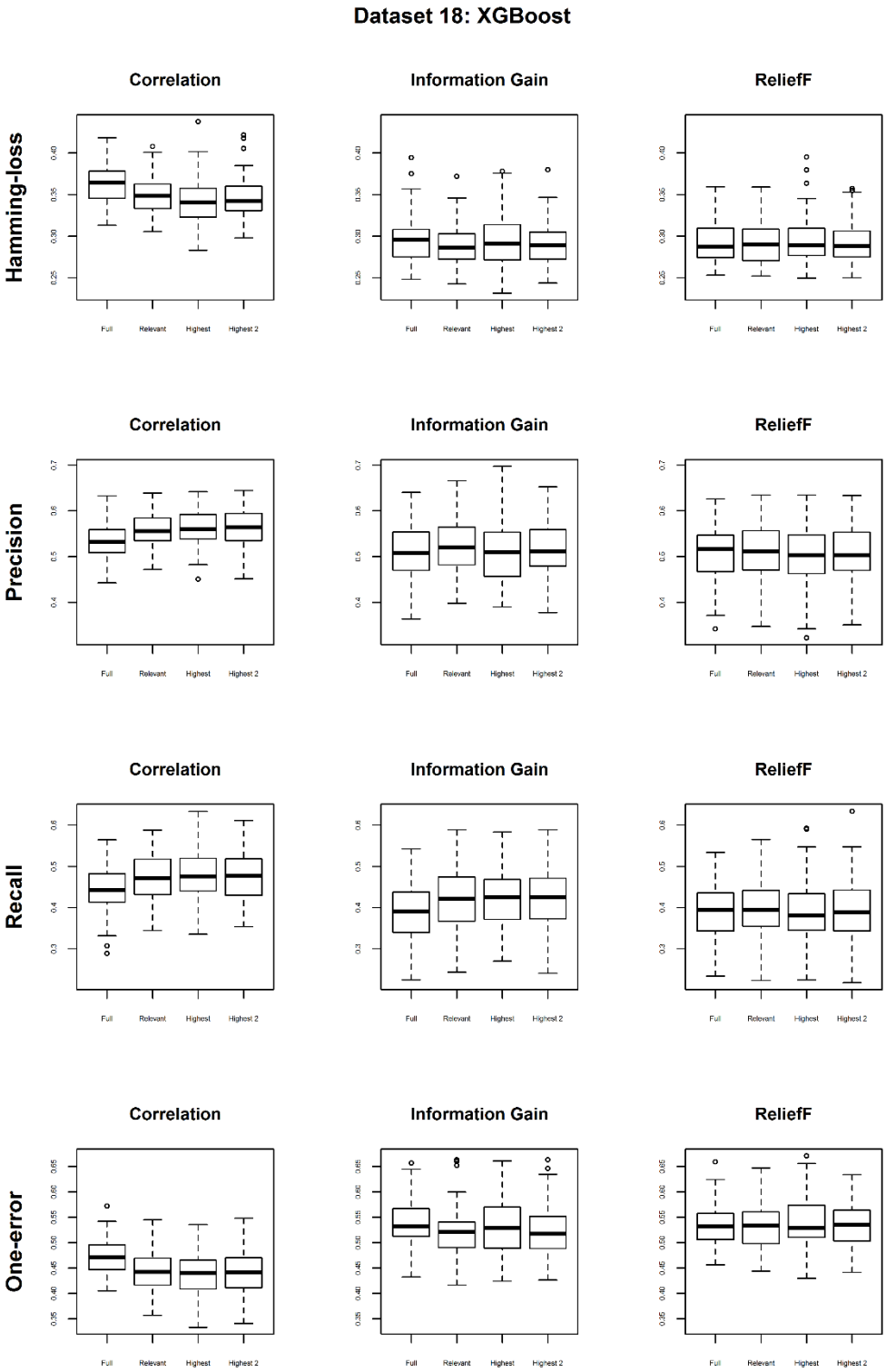


Figure E.18 Summary of results for the XGBoost classifier: Dataset 18.

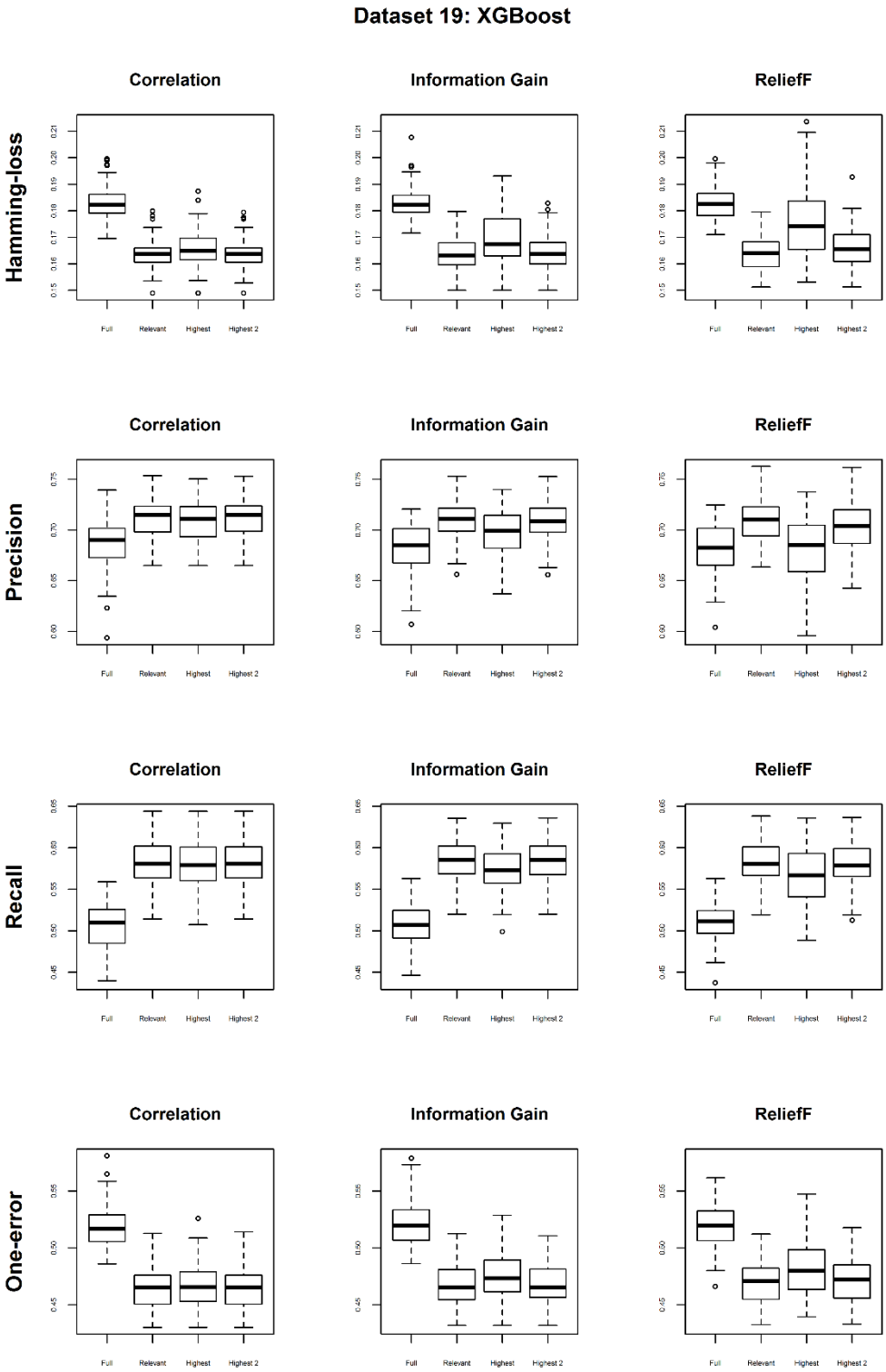


Figure E.19 Summary of results for the XGBoost classifier: Dataset 19.

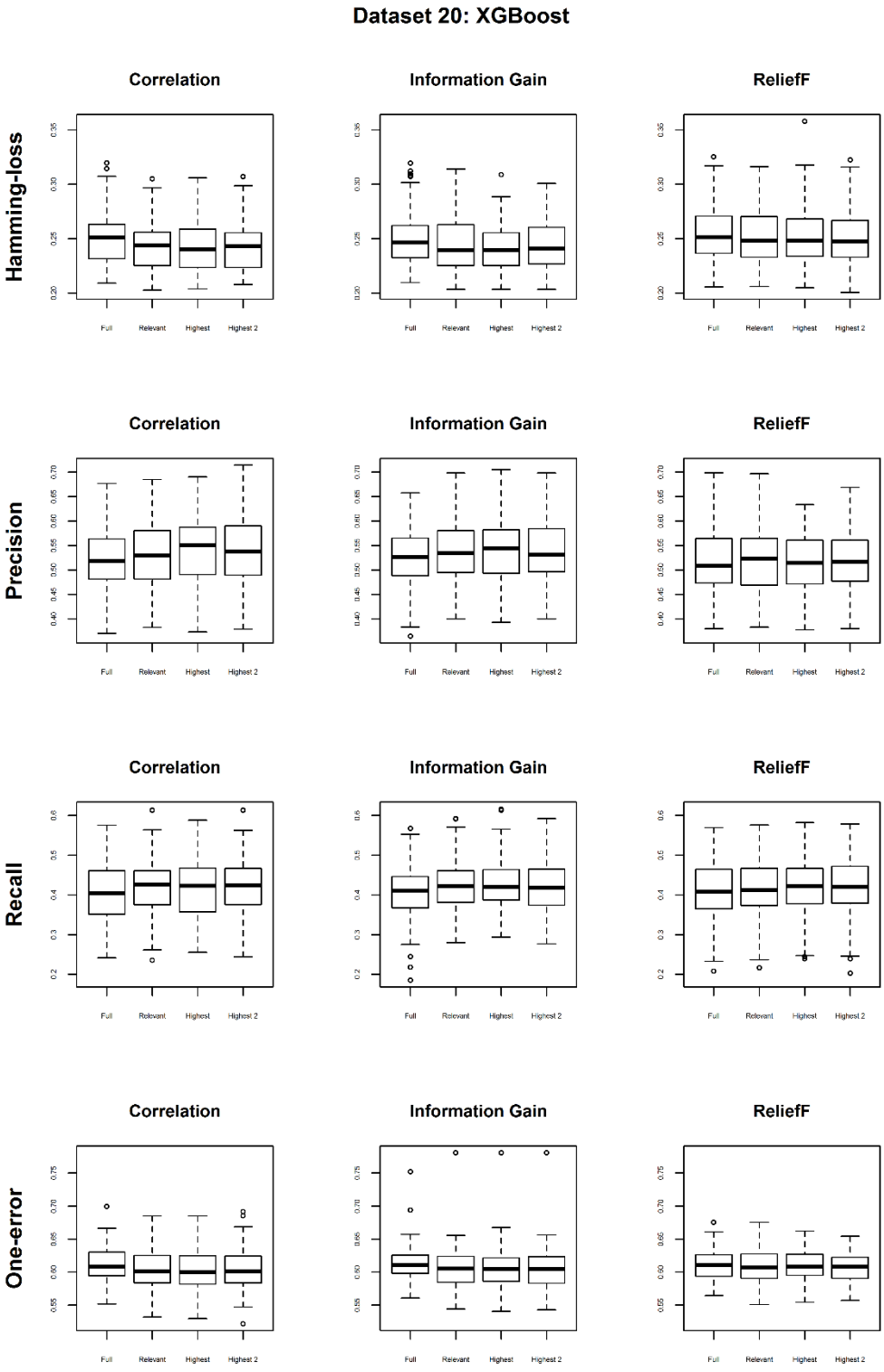


Figure E.20 Summary of results for the XGBoost classifier: Dataset 20.

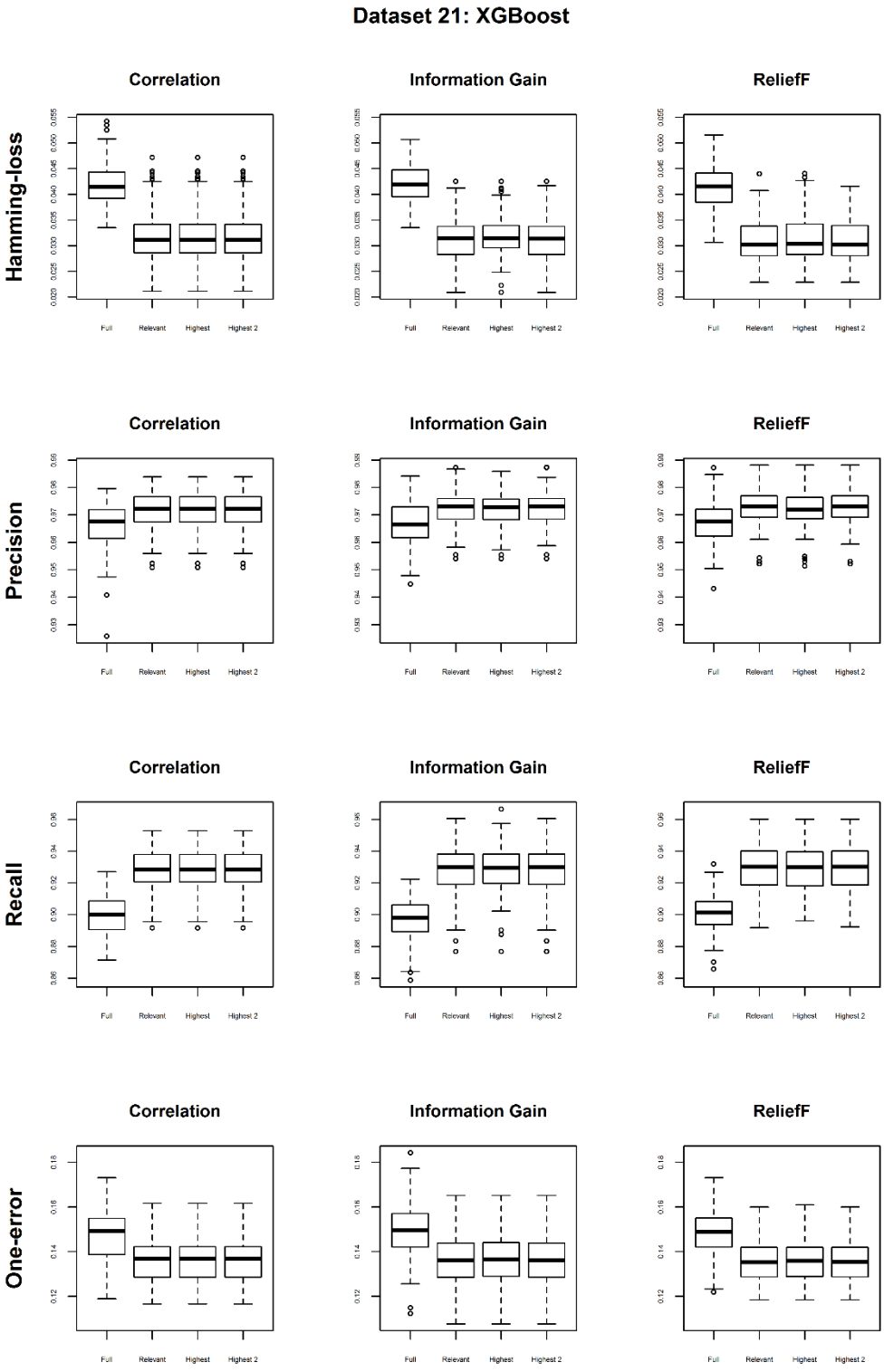


Figure E.21 Summary of results for the XGBoost classifier: Dataset 21.

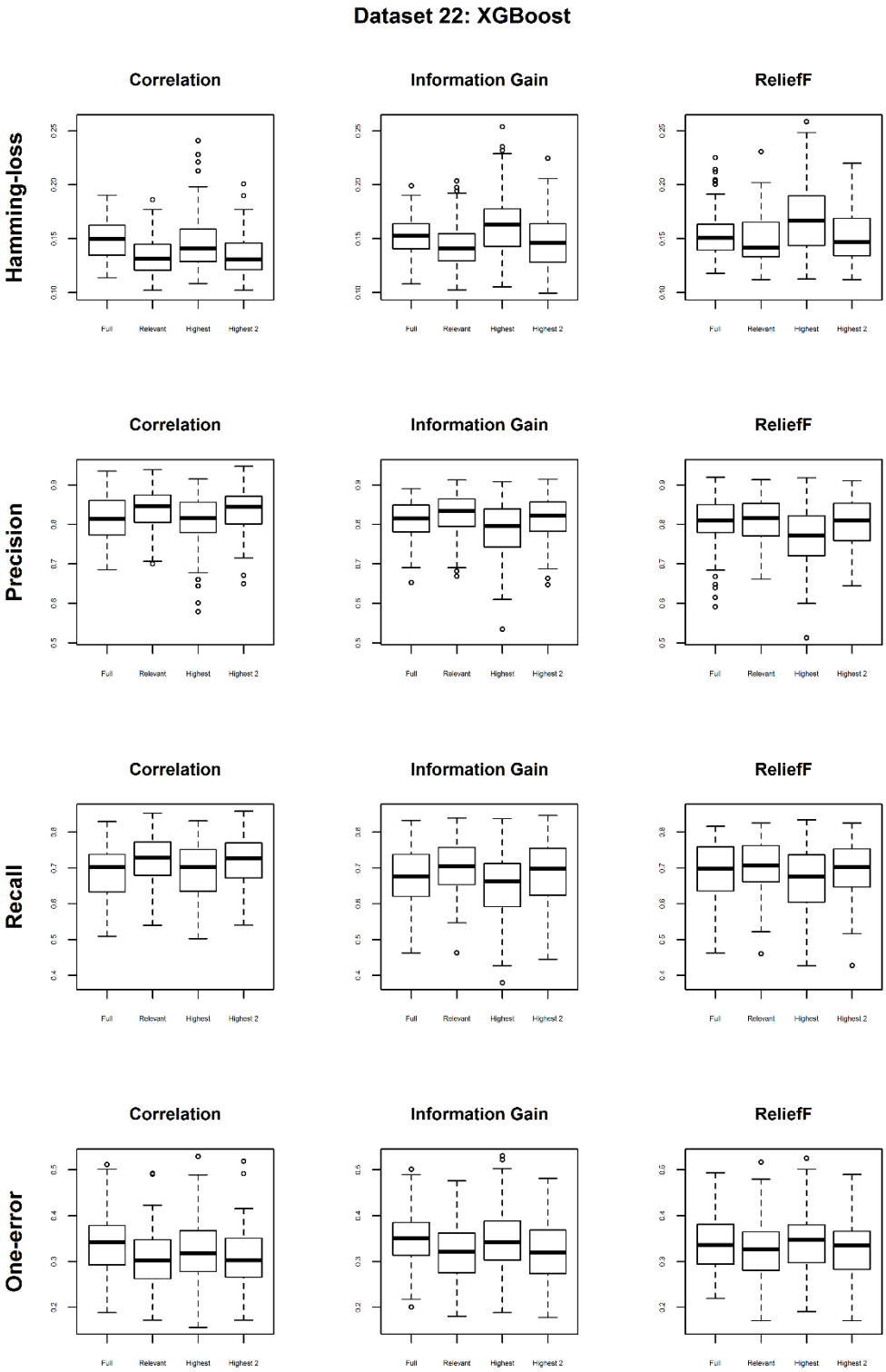


Figure E.22 Summary of results for the XGBoost classifier: Dataset 22.

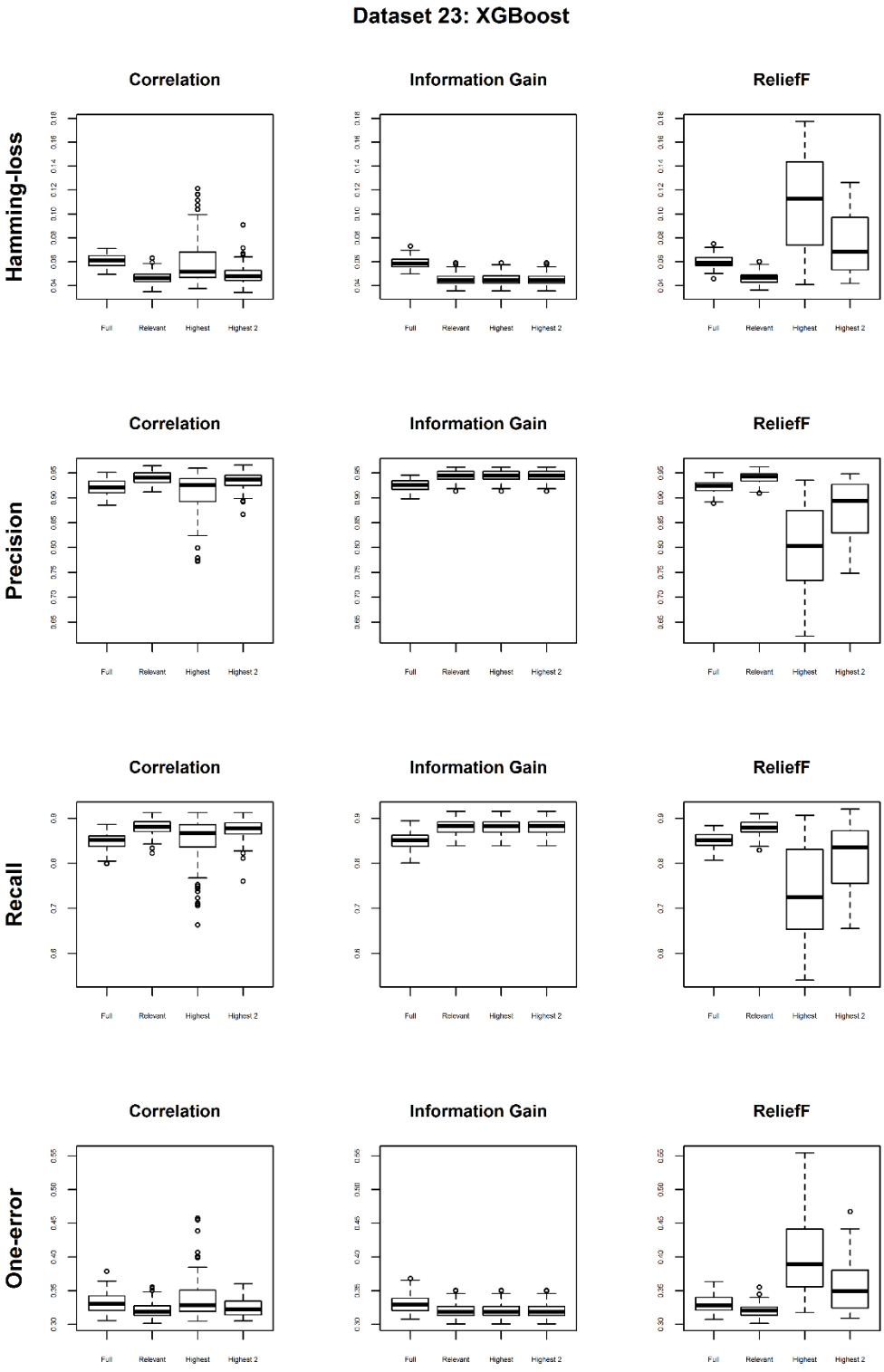


Figure E.23 Summary of results for the XGBoost classifier: Dataset 23.

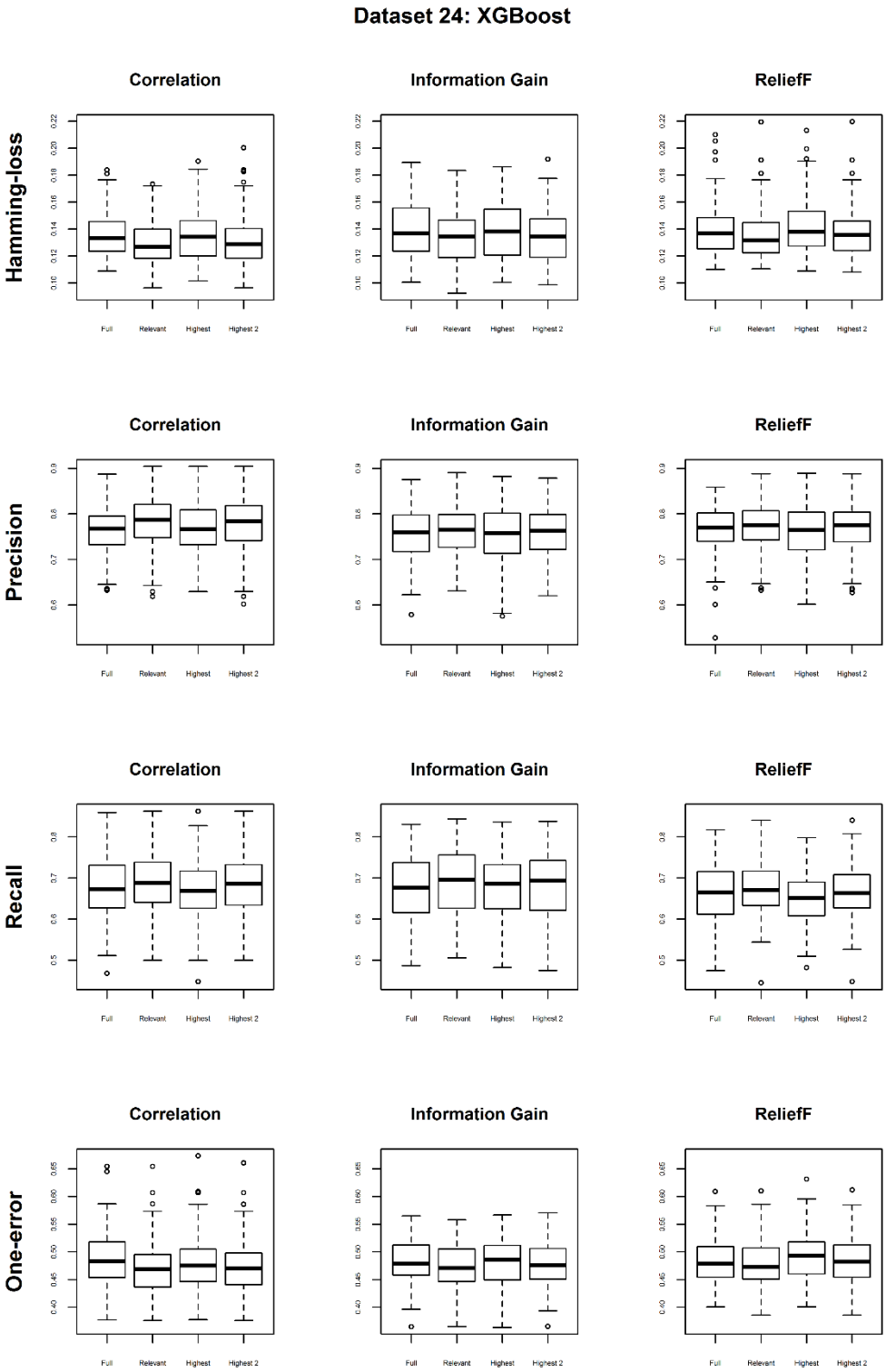


Figure E.24 Summary of results for the XGBoost classifier: Dataset 24.

APPENDIX F

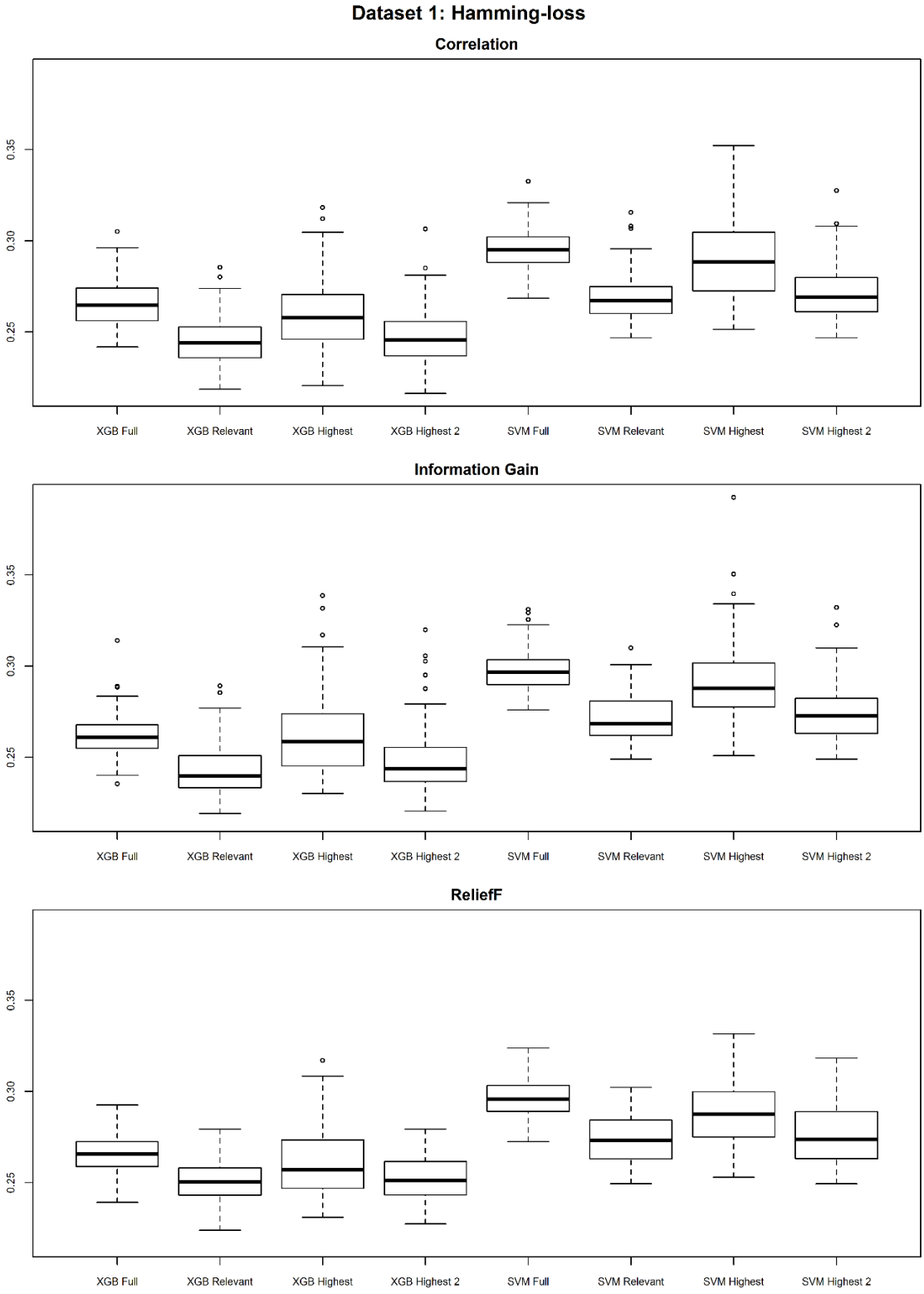


Figure F.1 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 1.

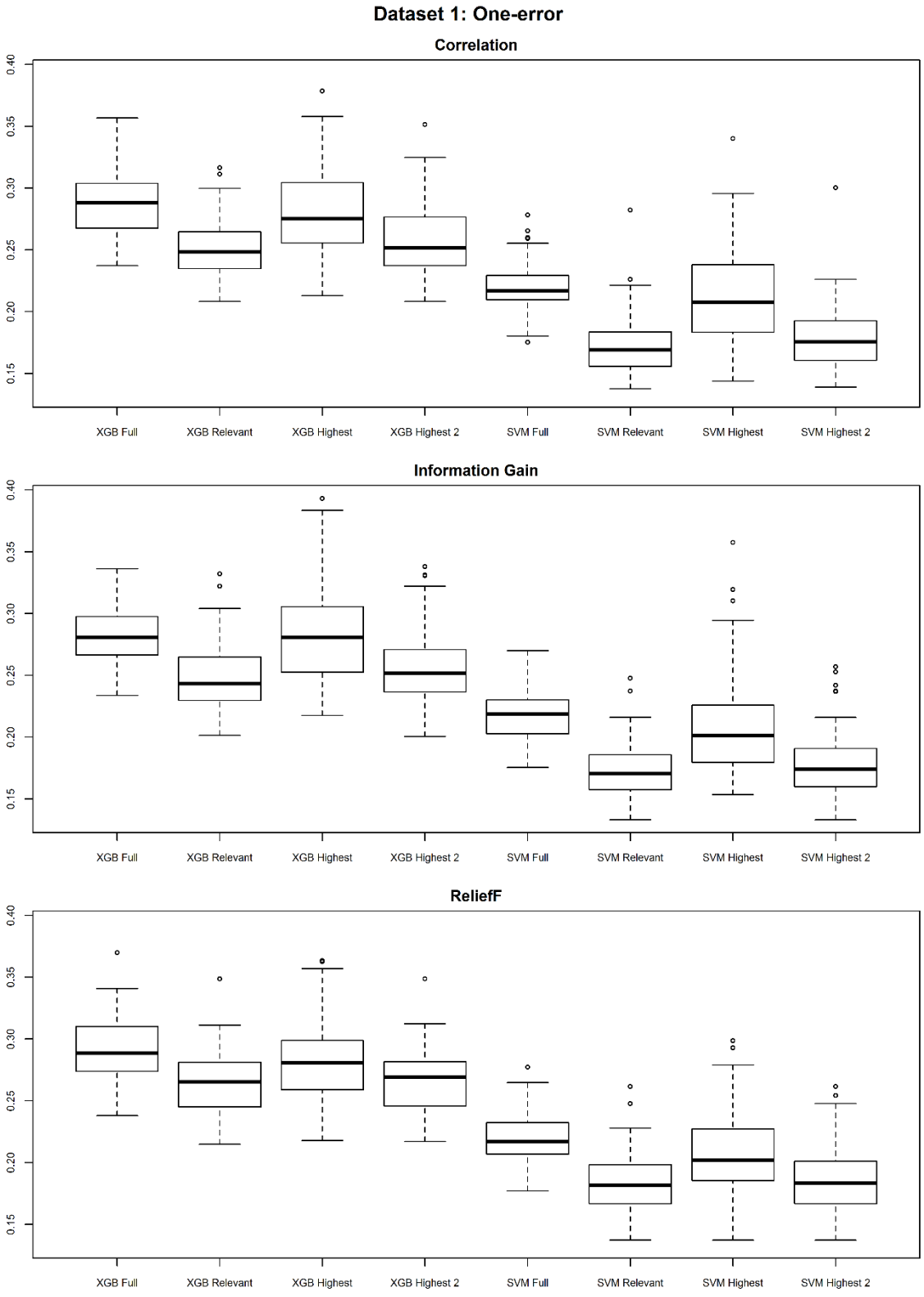


Figure F.2 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 1.

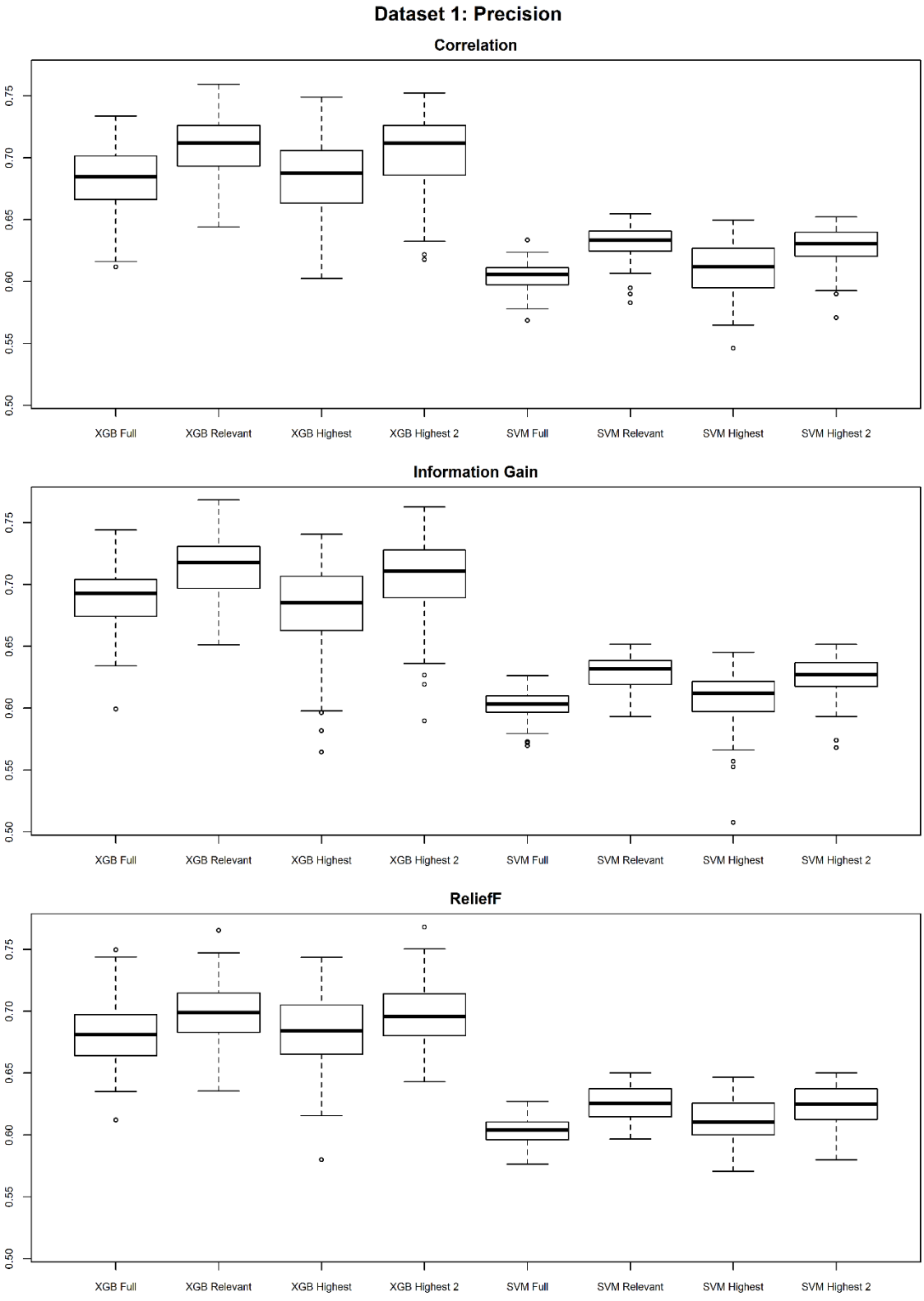


Figure F.3 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 1.

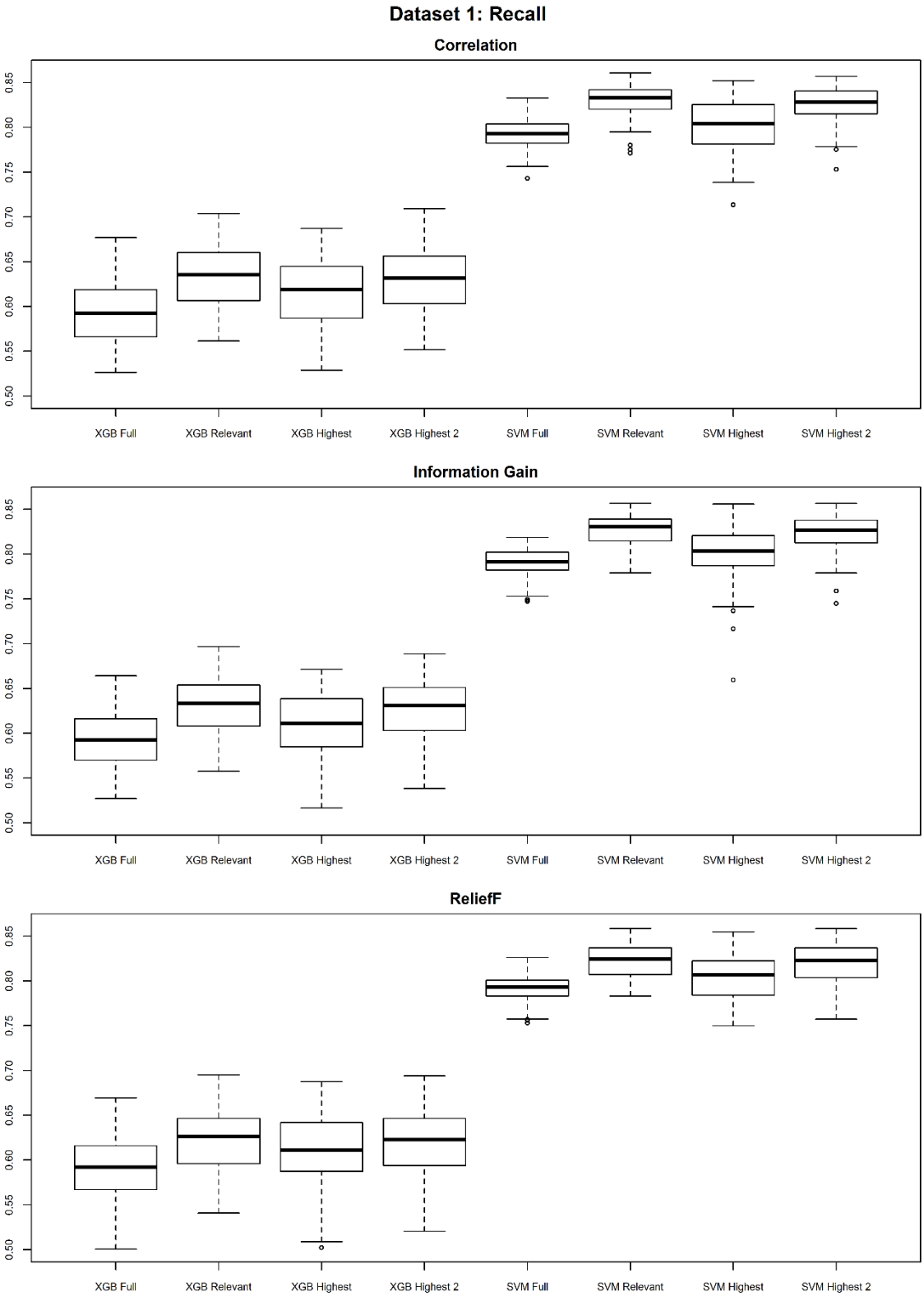


Figure F.4 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 1.

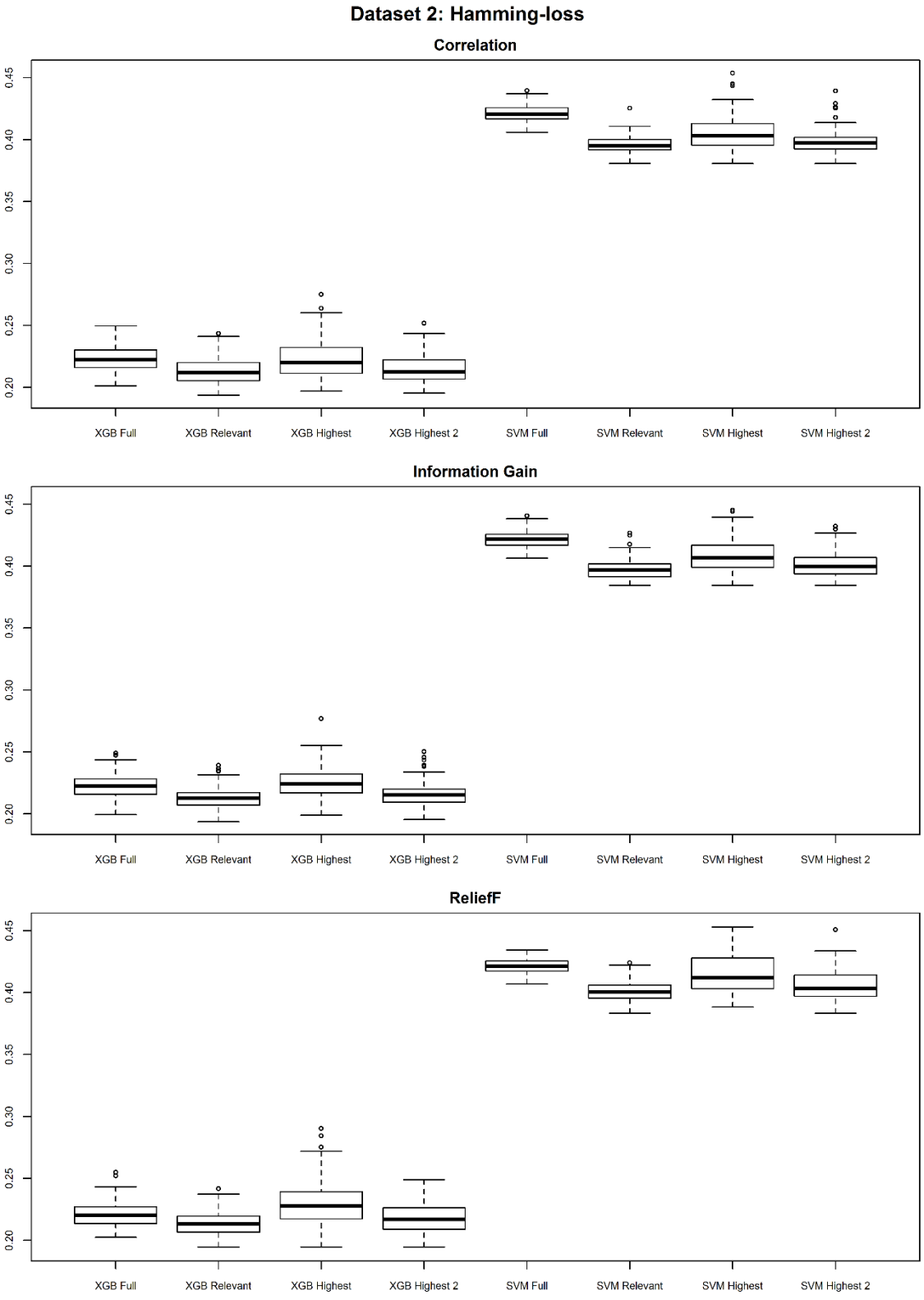


Figure F.5 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 2.

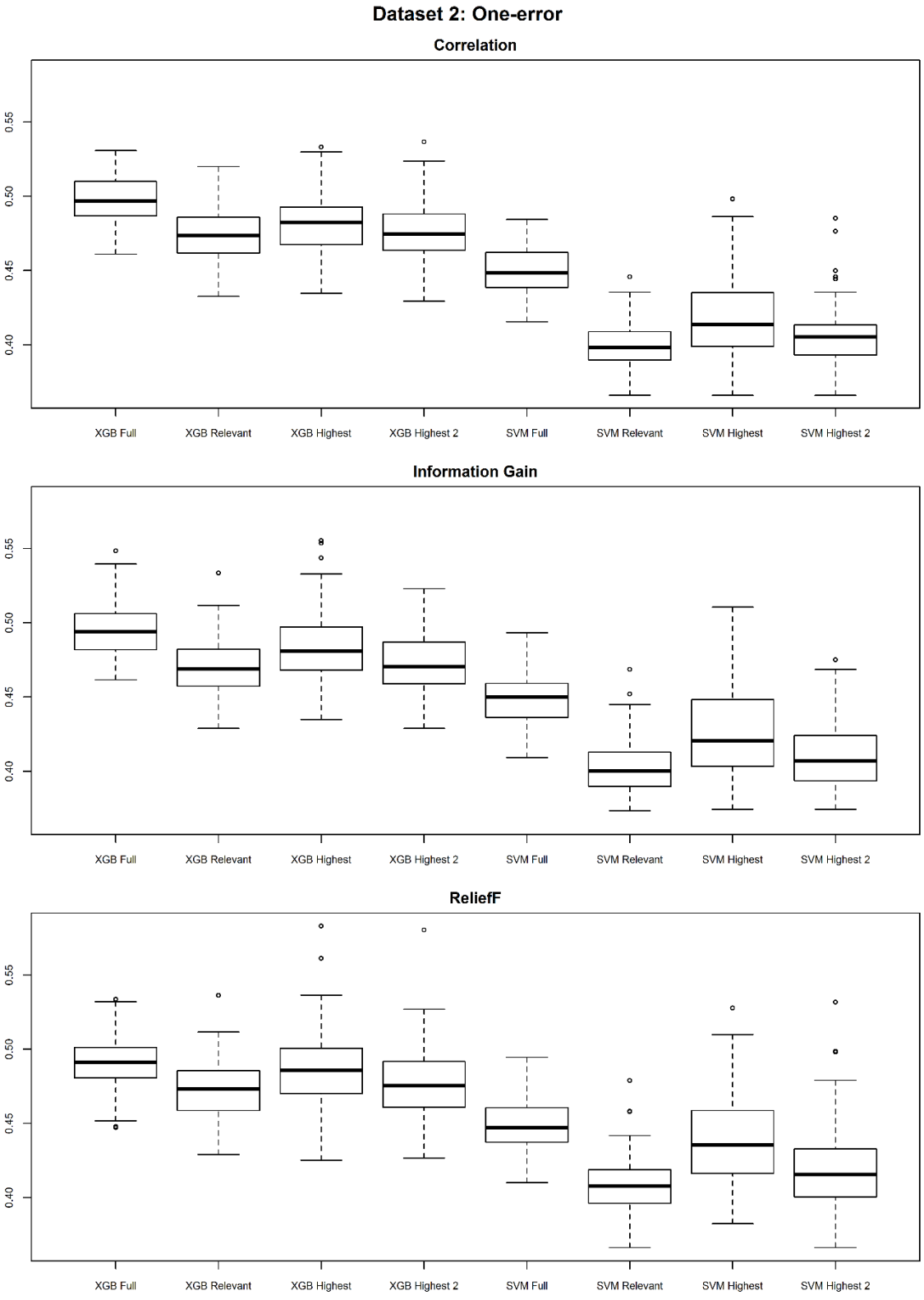


Figure F.6 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 2.

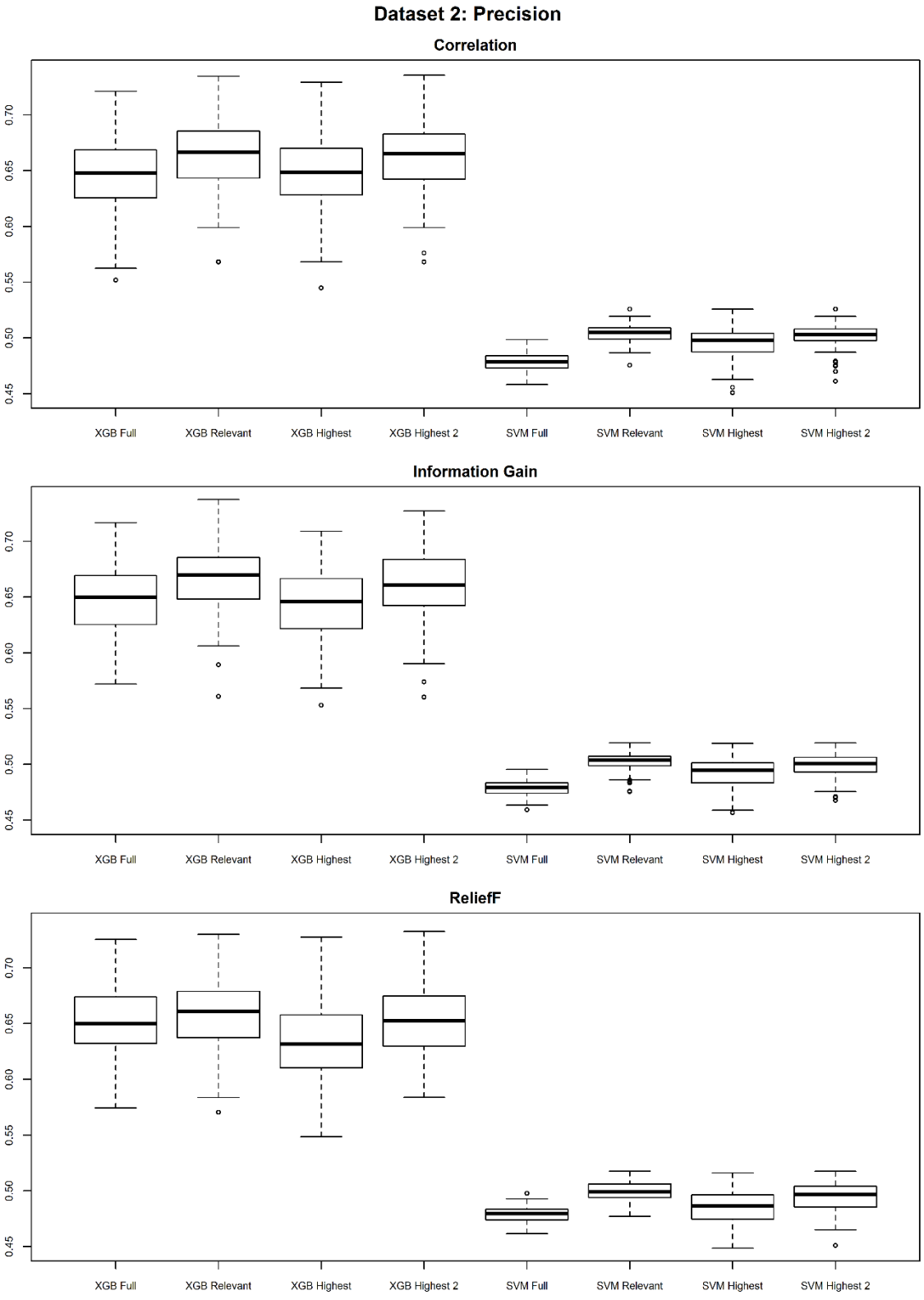


Figure F.7 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 2.

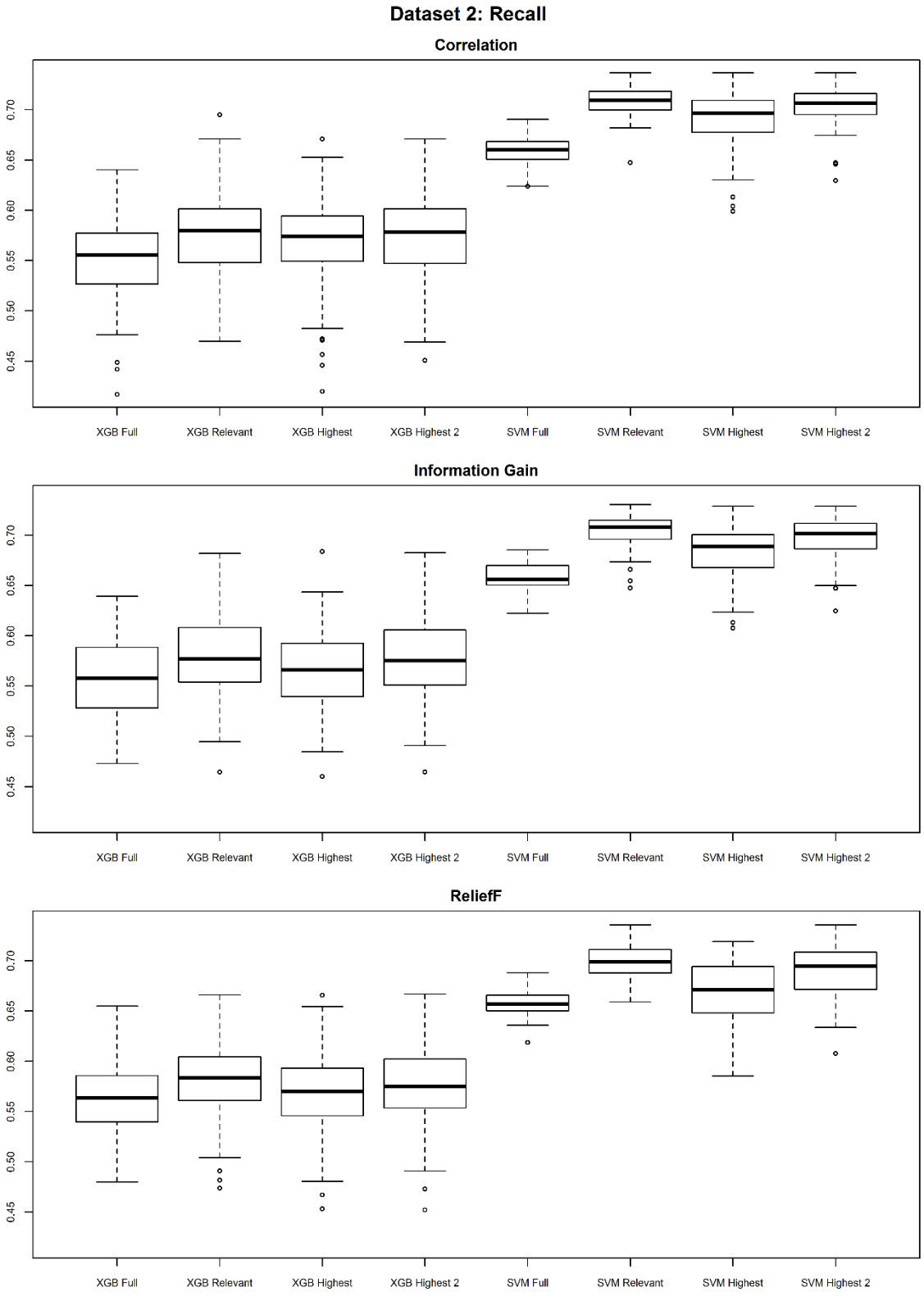


Figure F.8 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 2.

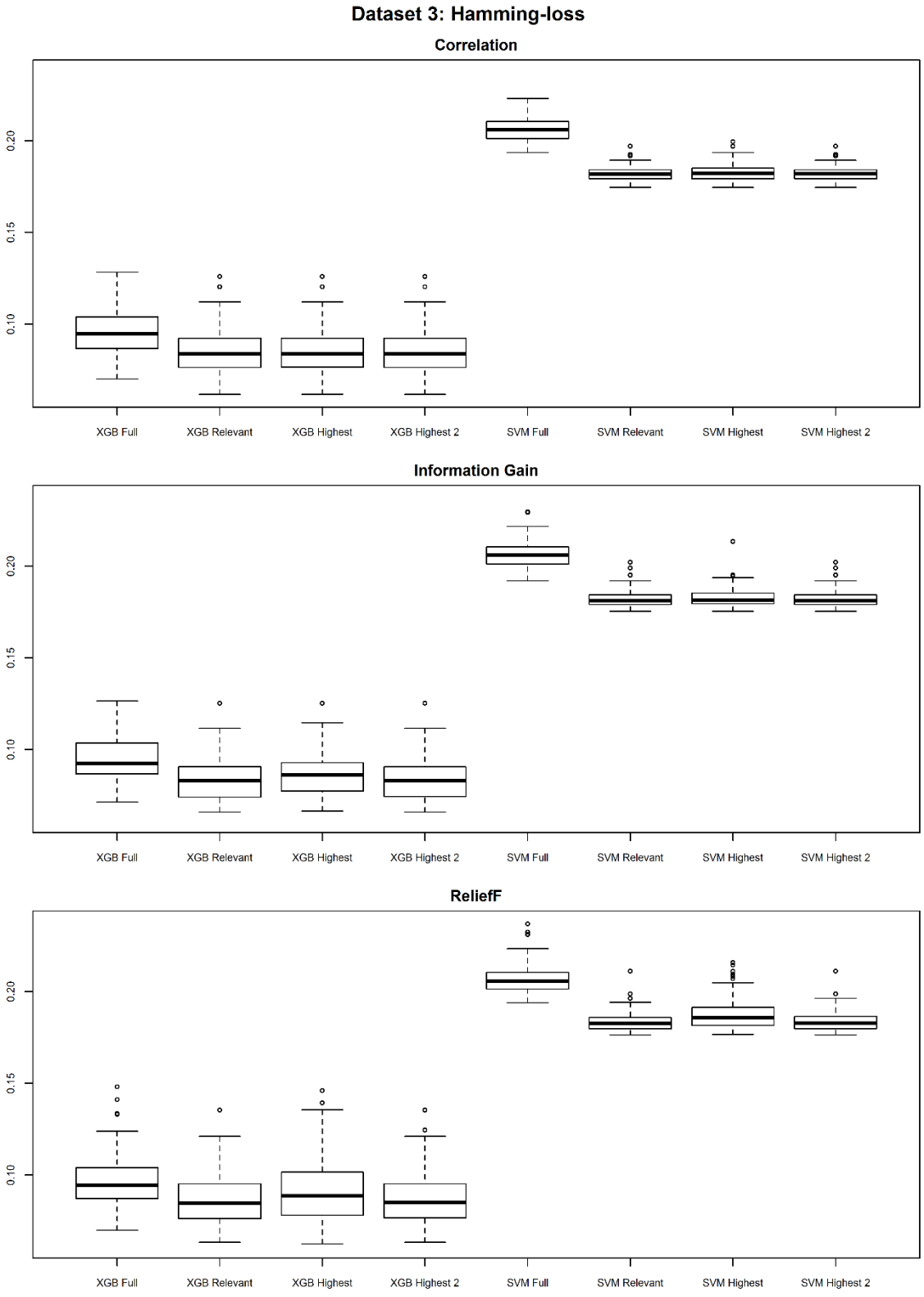


Figure F.9 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 3.

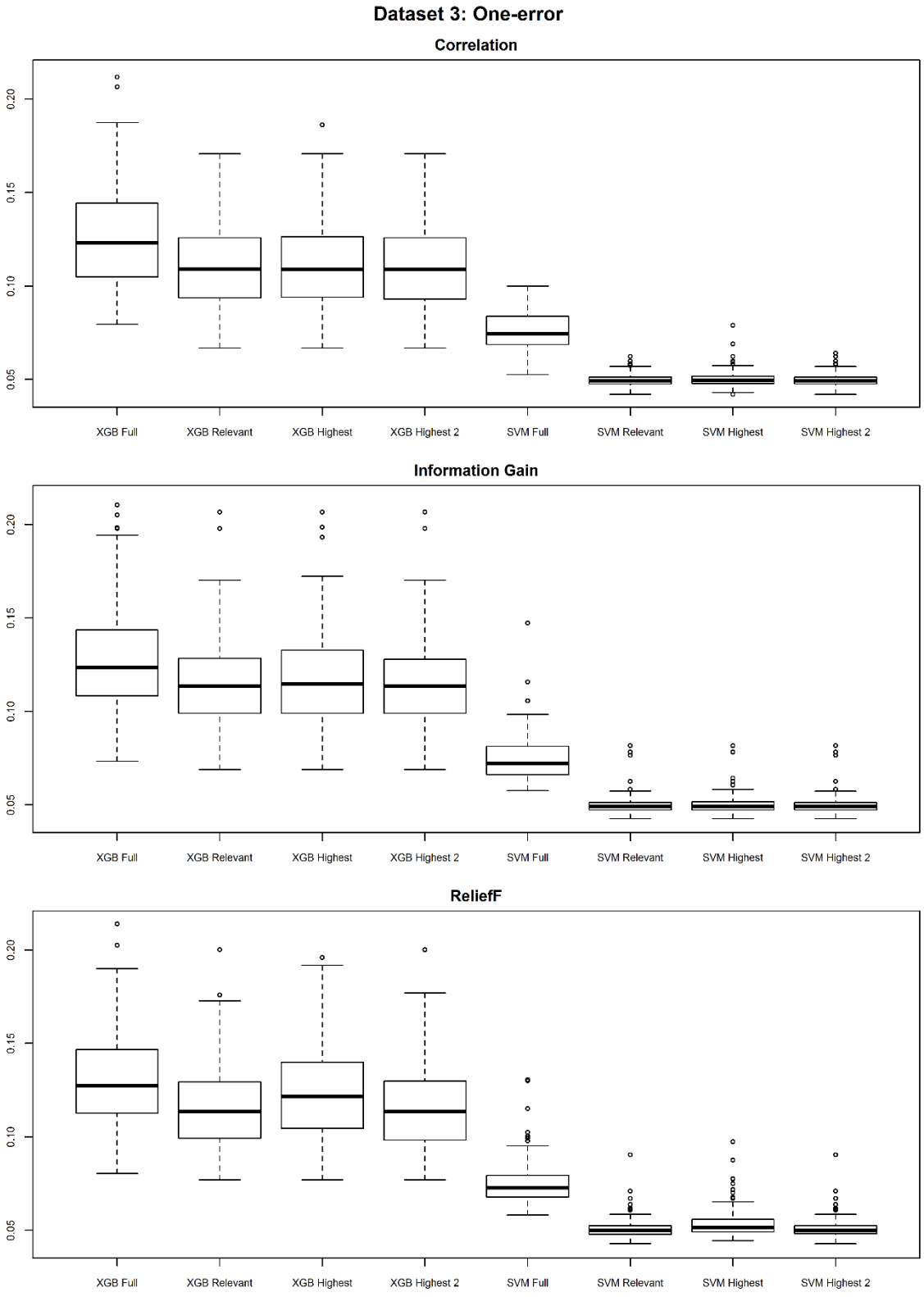


Figure F.10 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 3.

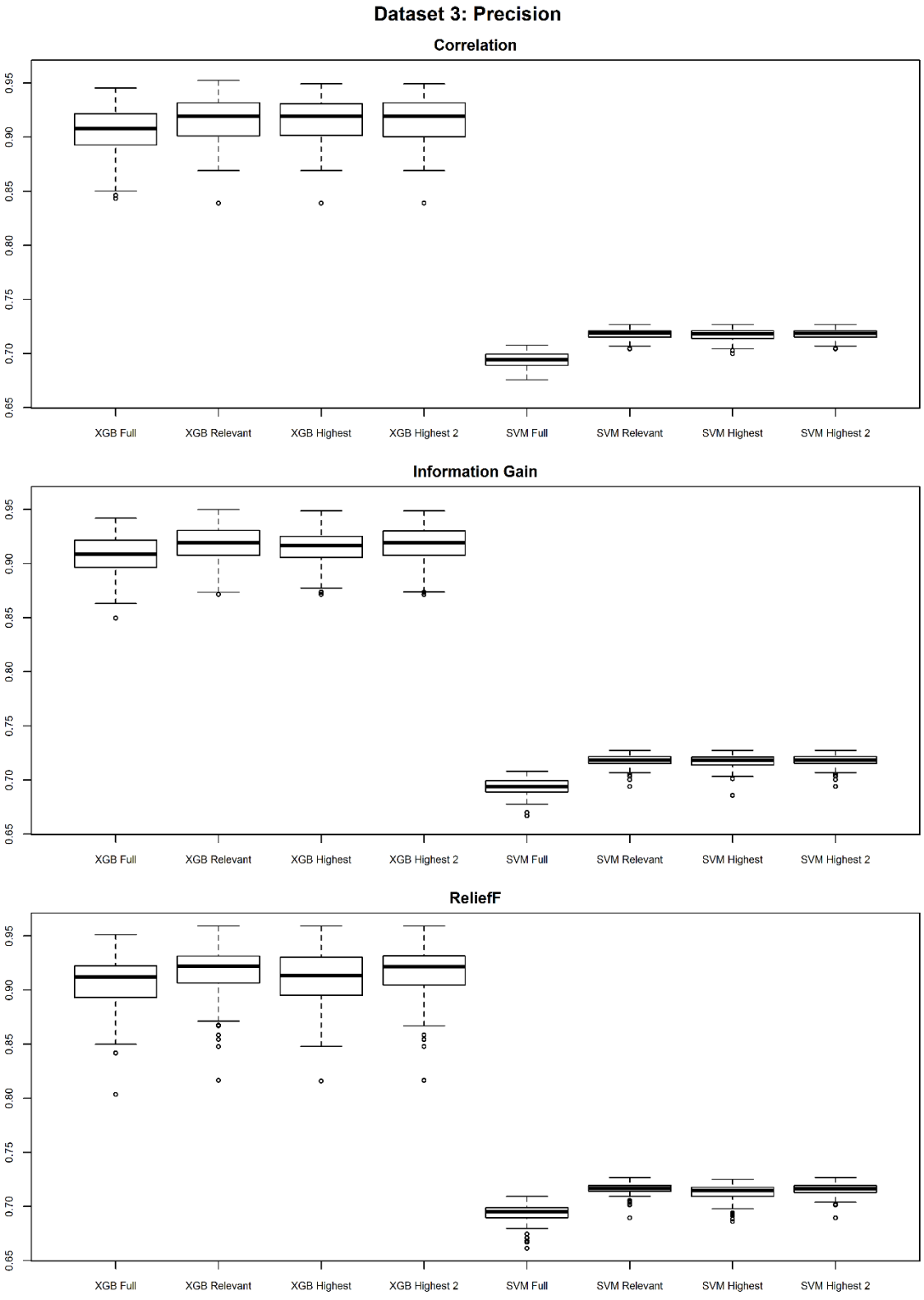


Figure F.11 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 3.

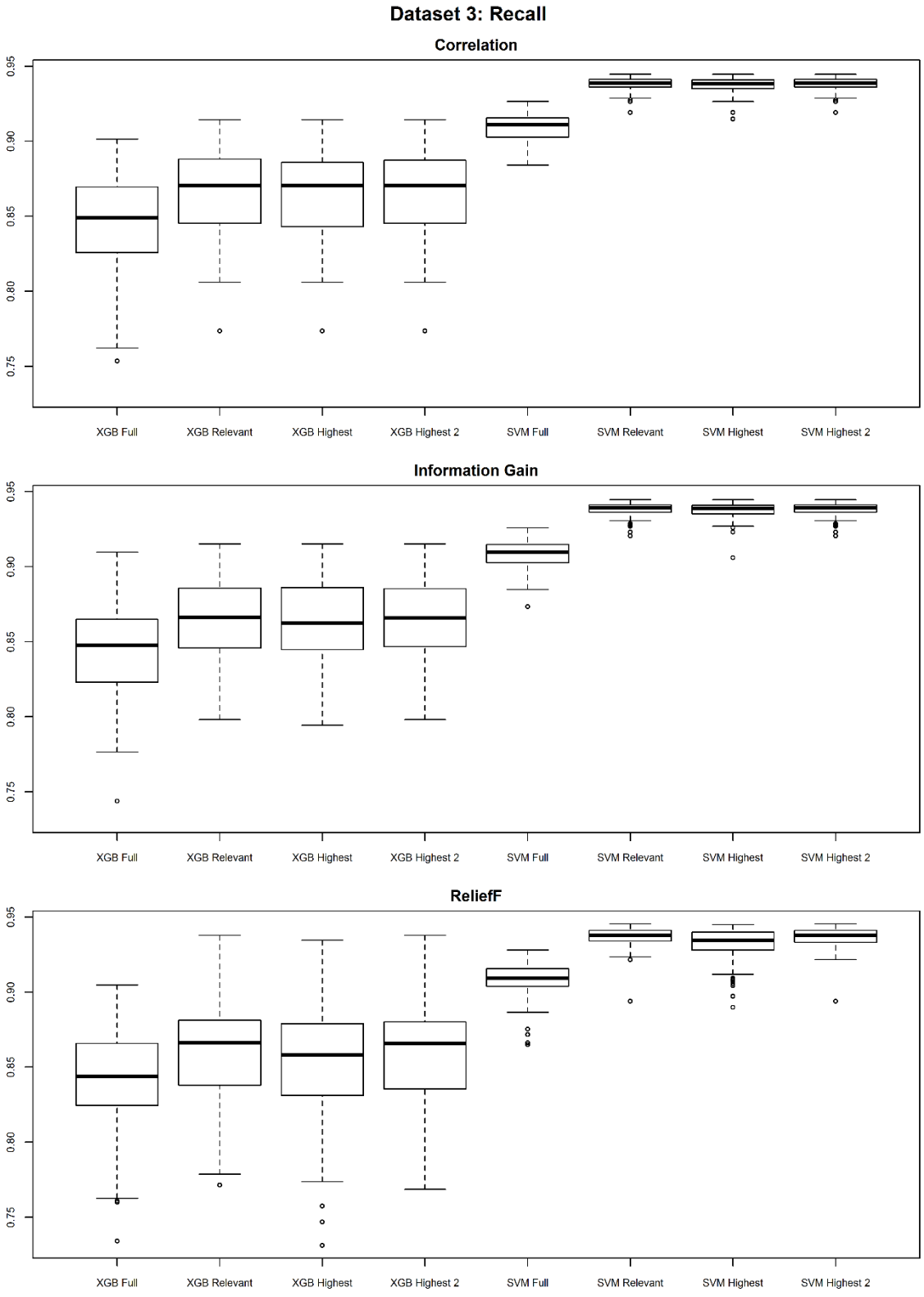


Figure F.12 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 3.

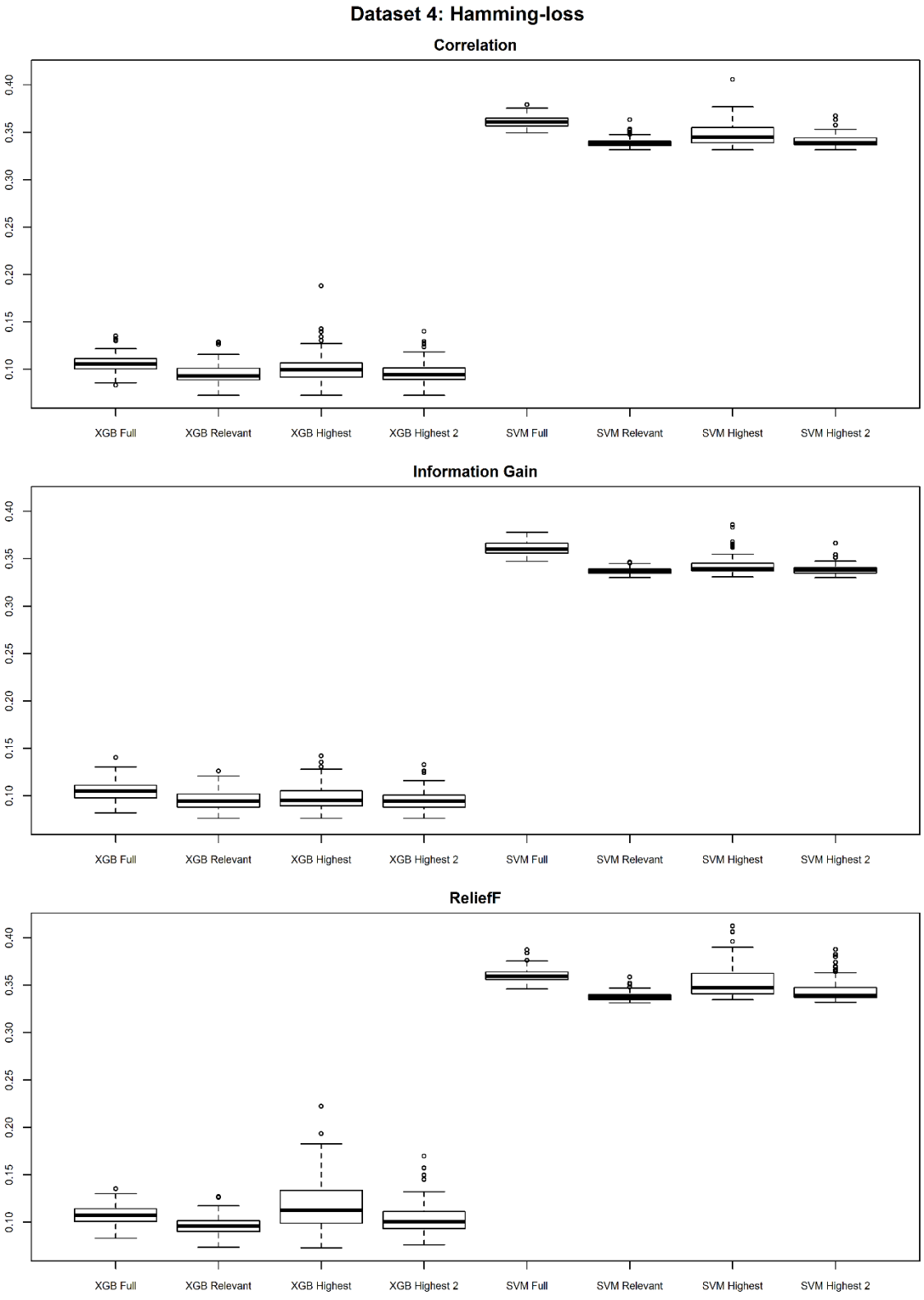


Figure F.13 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 4.

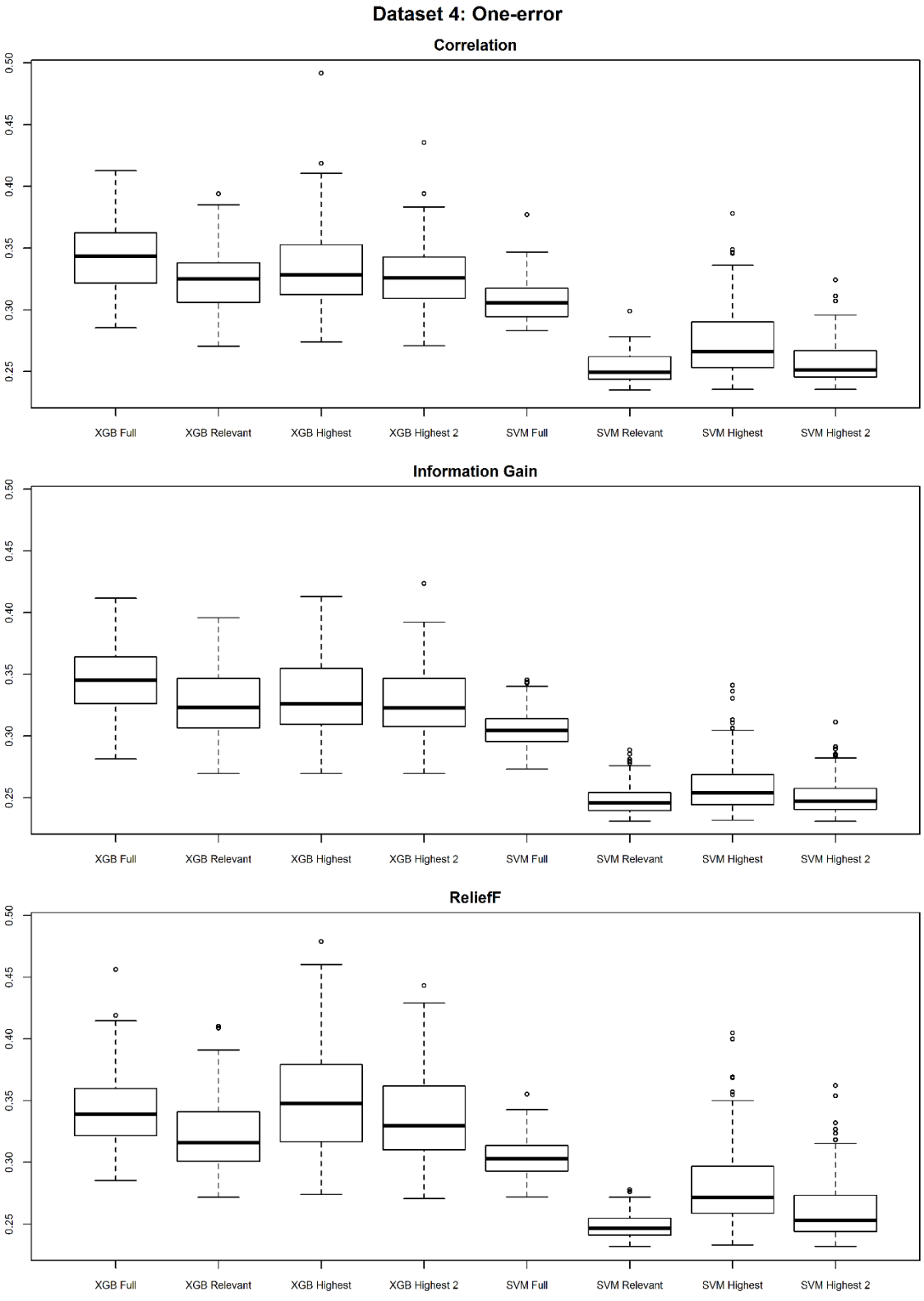


Figure F.14 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 4.

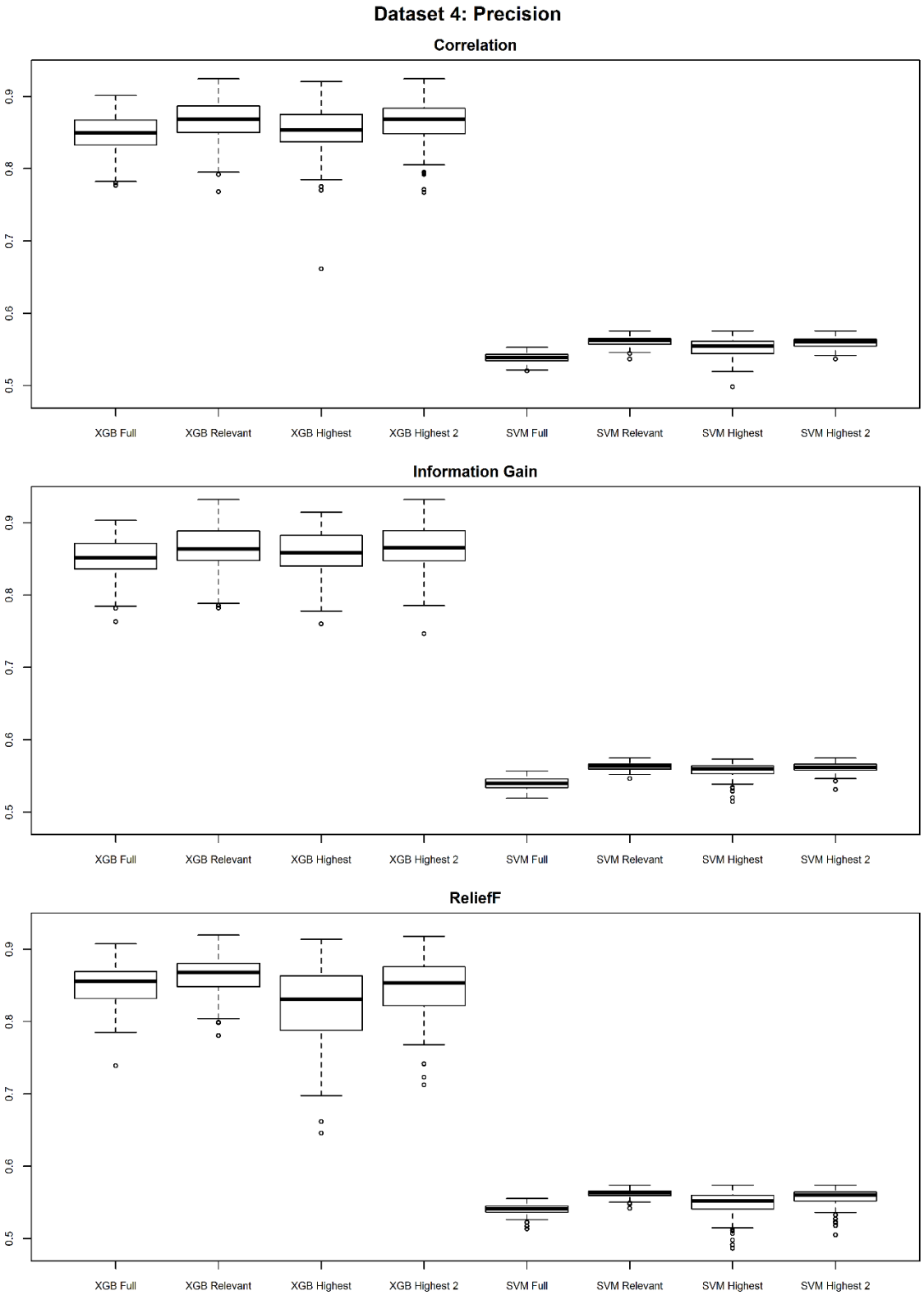


Figure F.15 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 4.

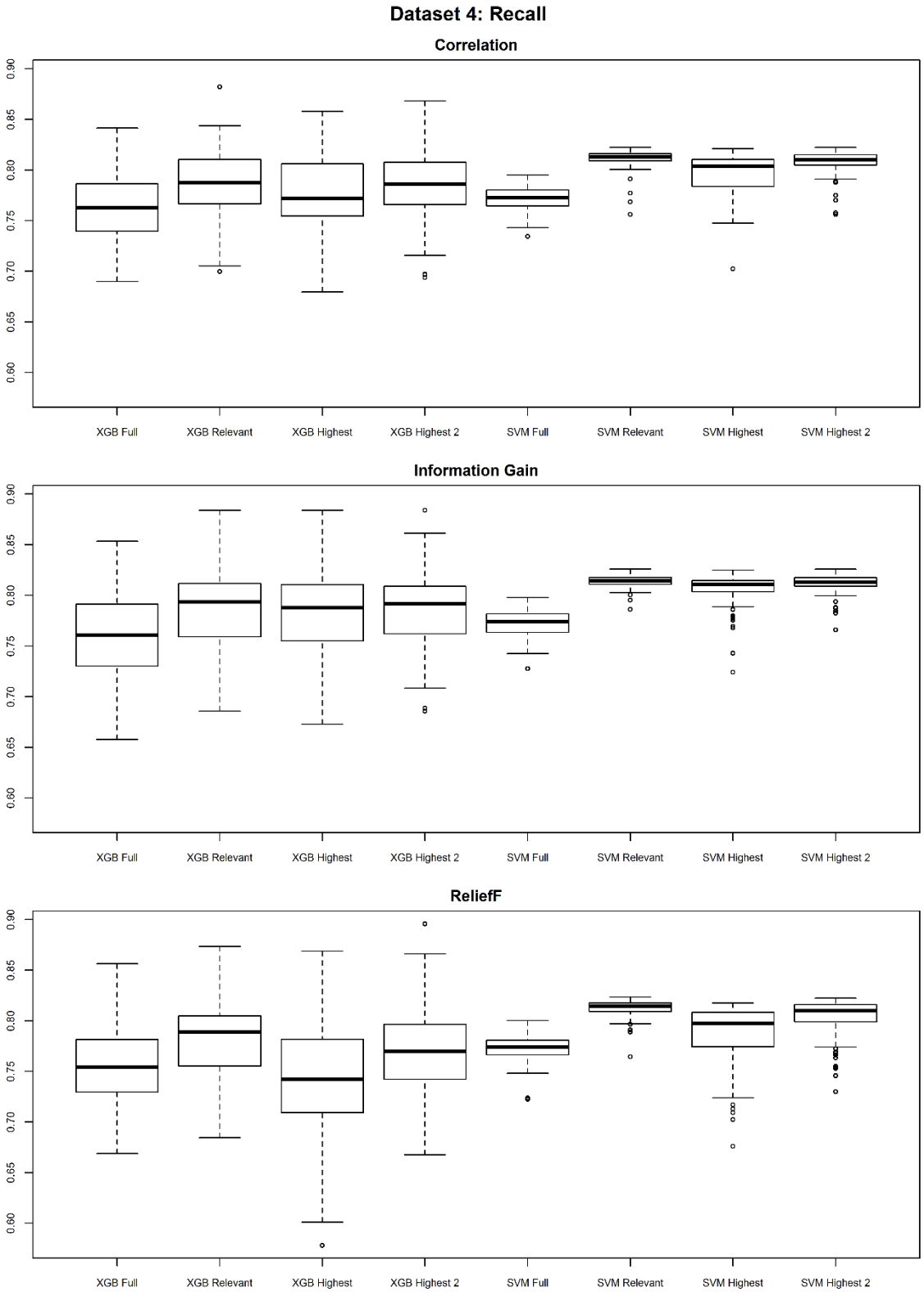


Figure F.16 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 4.

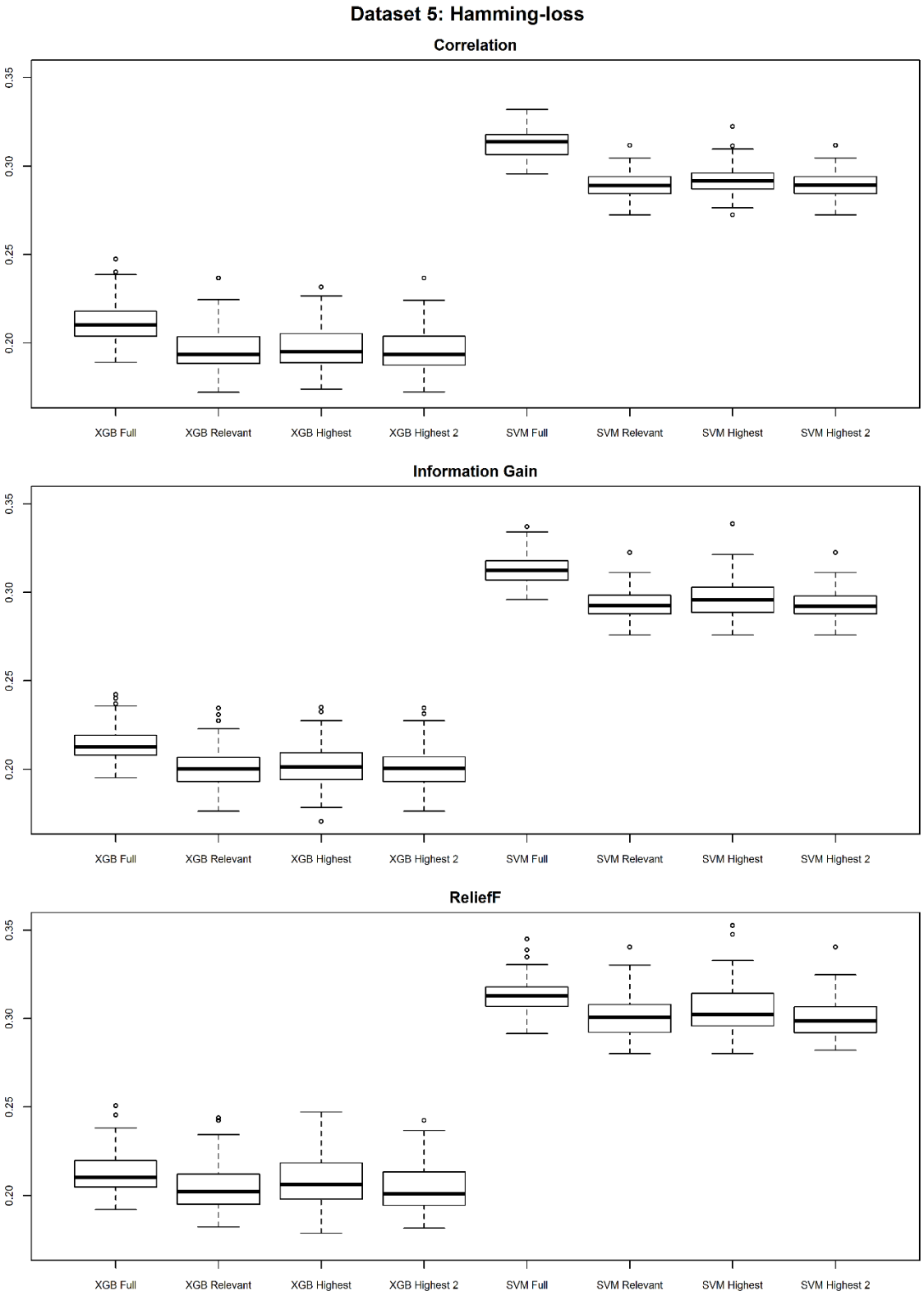


Figure F.17 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 5.

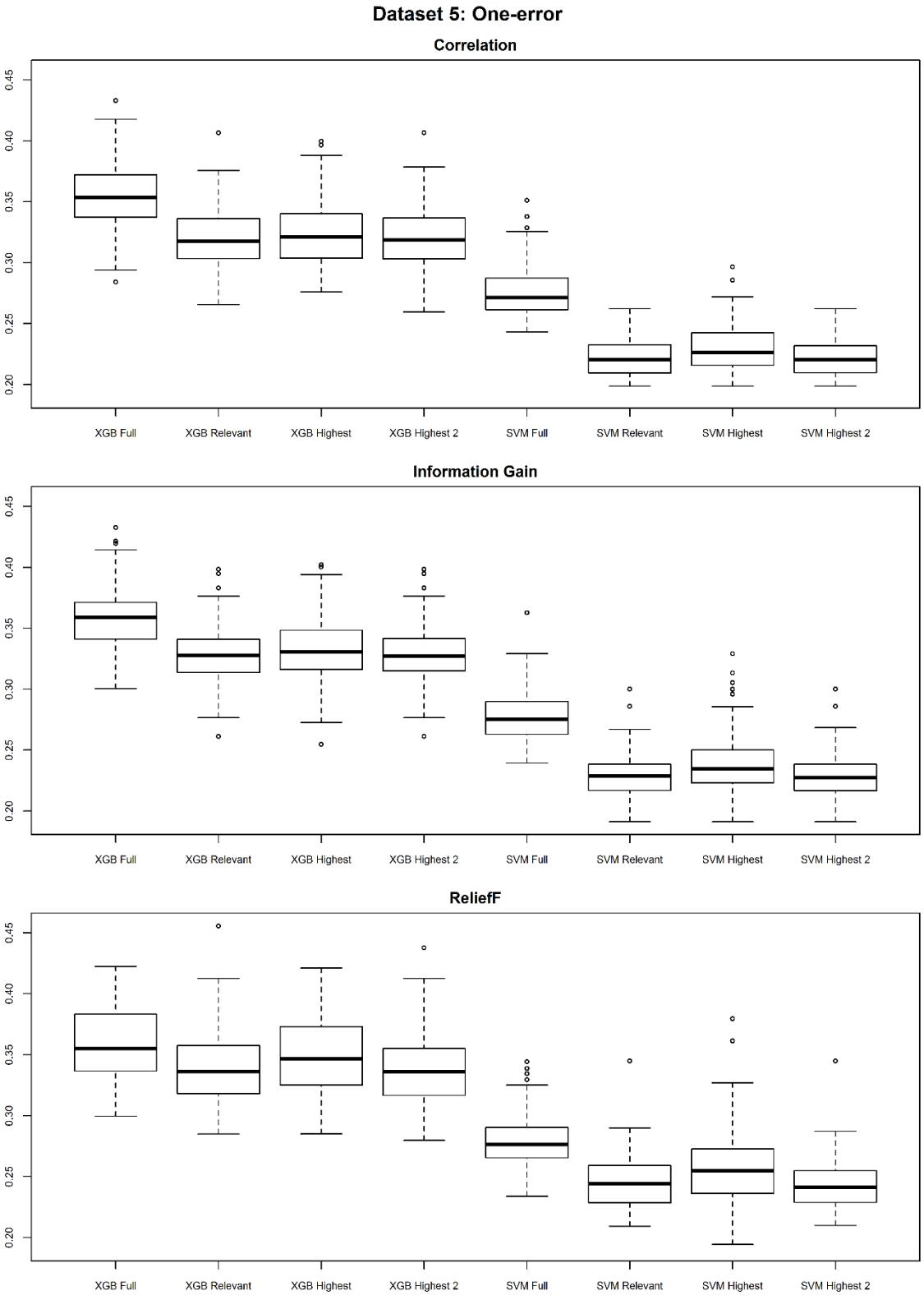


Figure F.18 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 5.

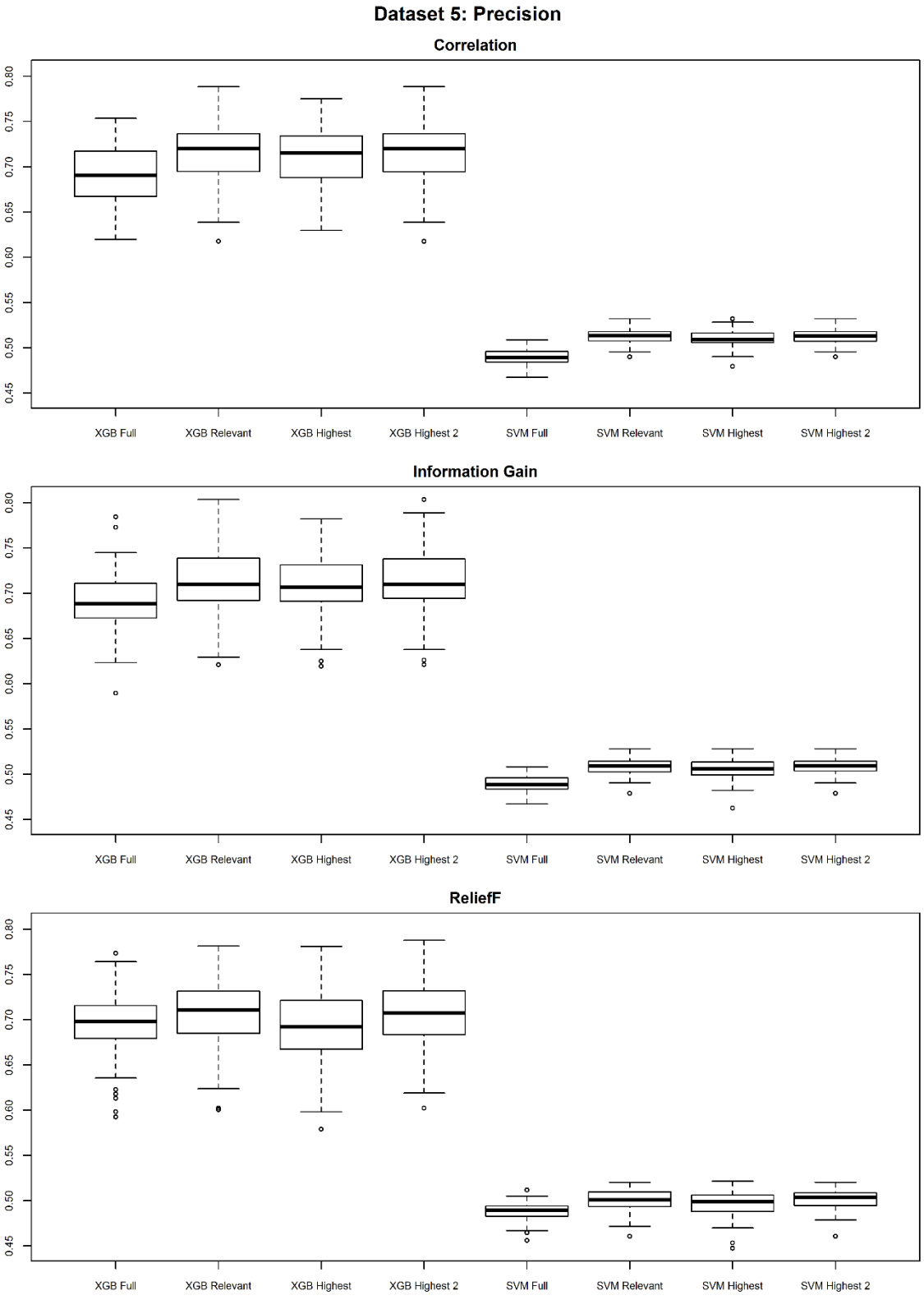


Figure F.19 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 5.

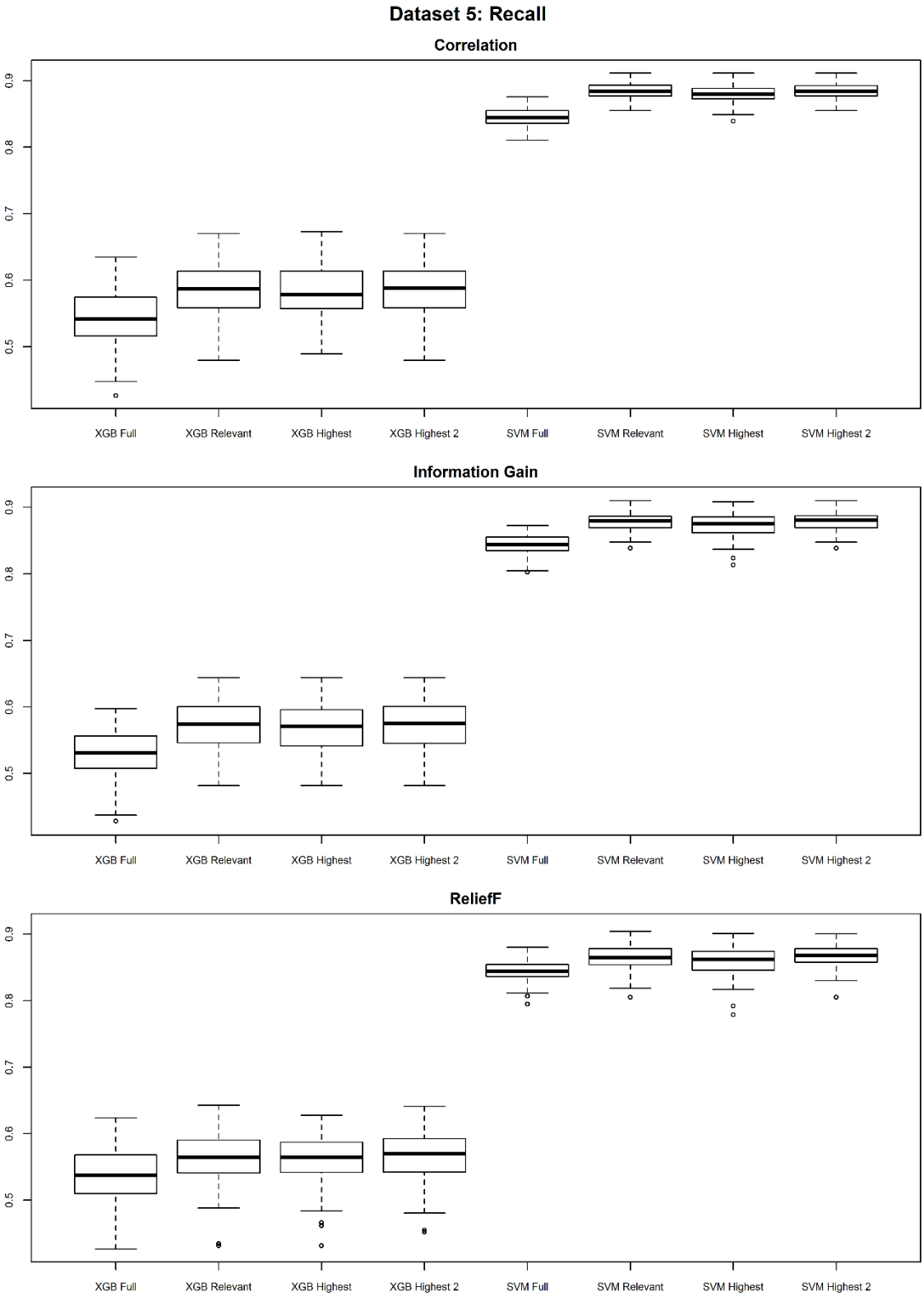


Figure F.20 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 5.

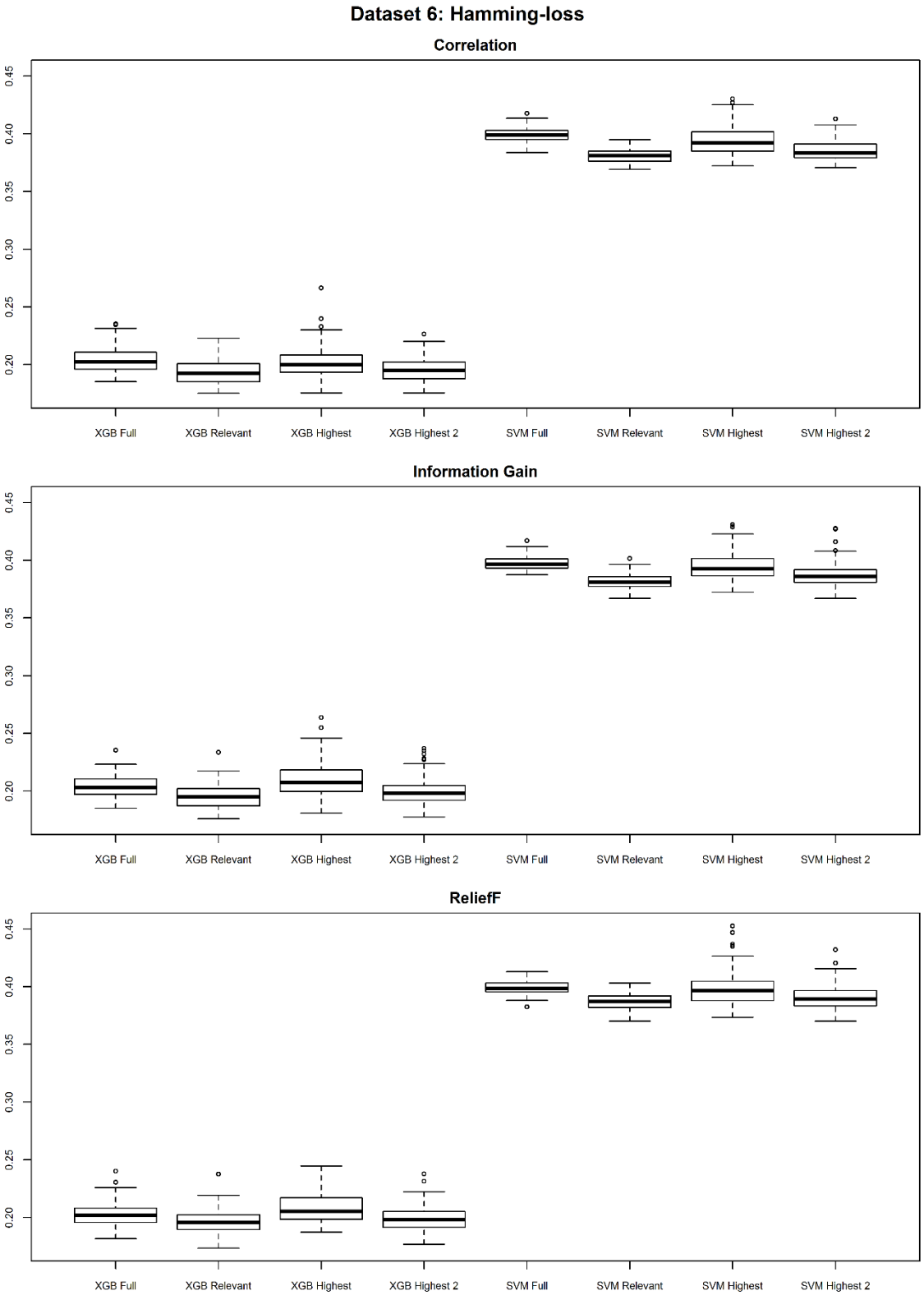


Figure F.21 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 6.

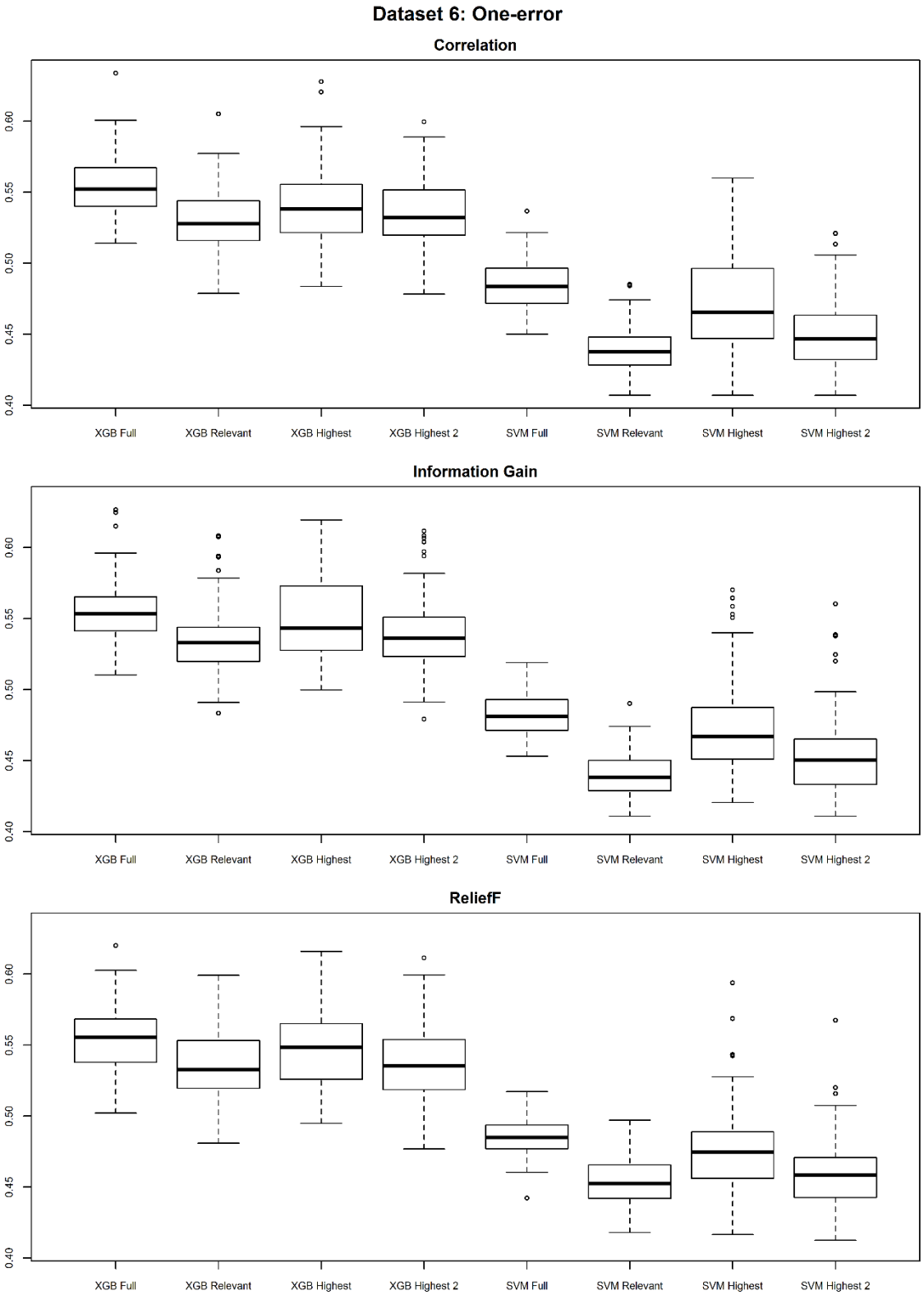


Figure F.22 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 6.

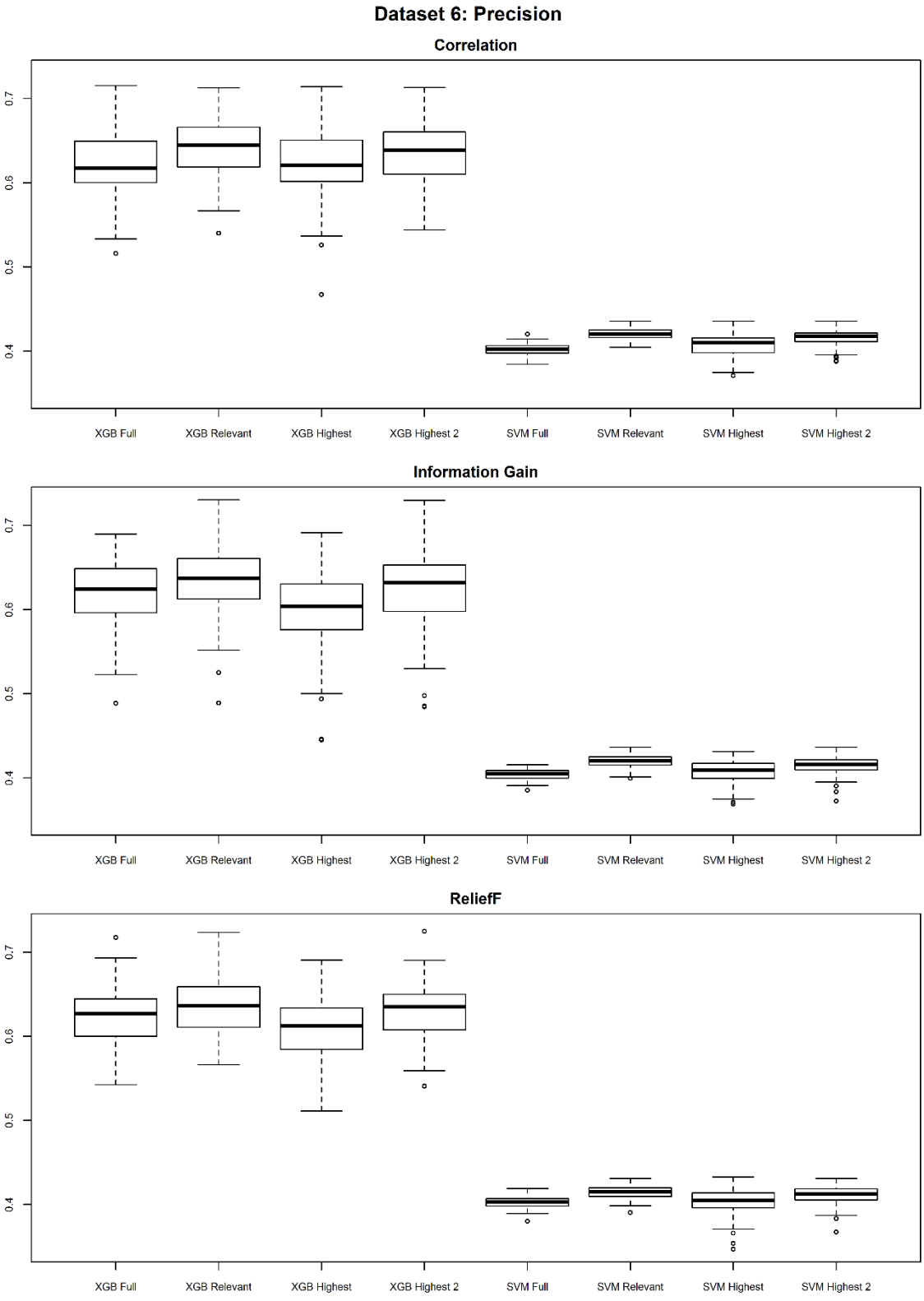


Figure F.23 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 6.

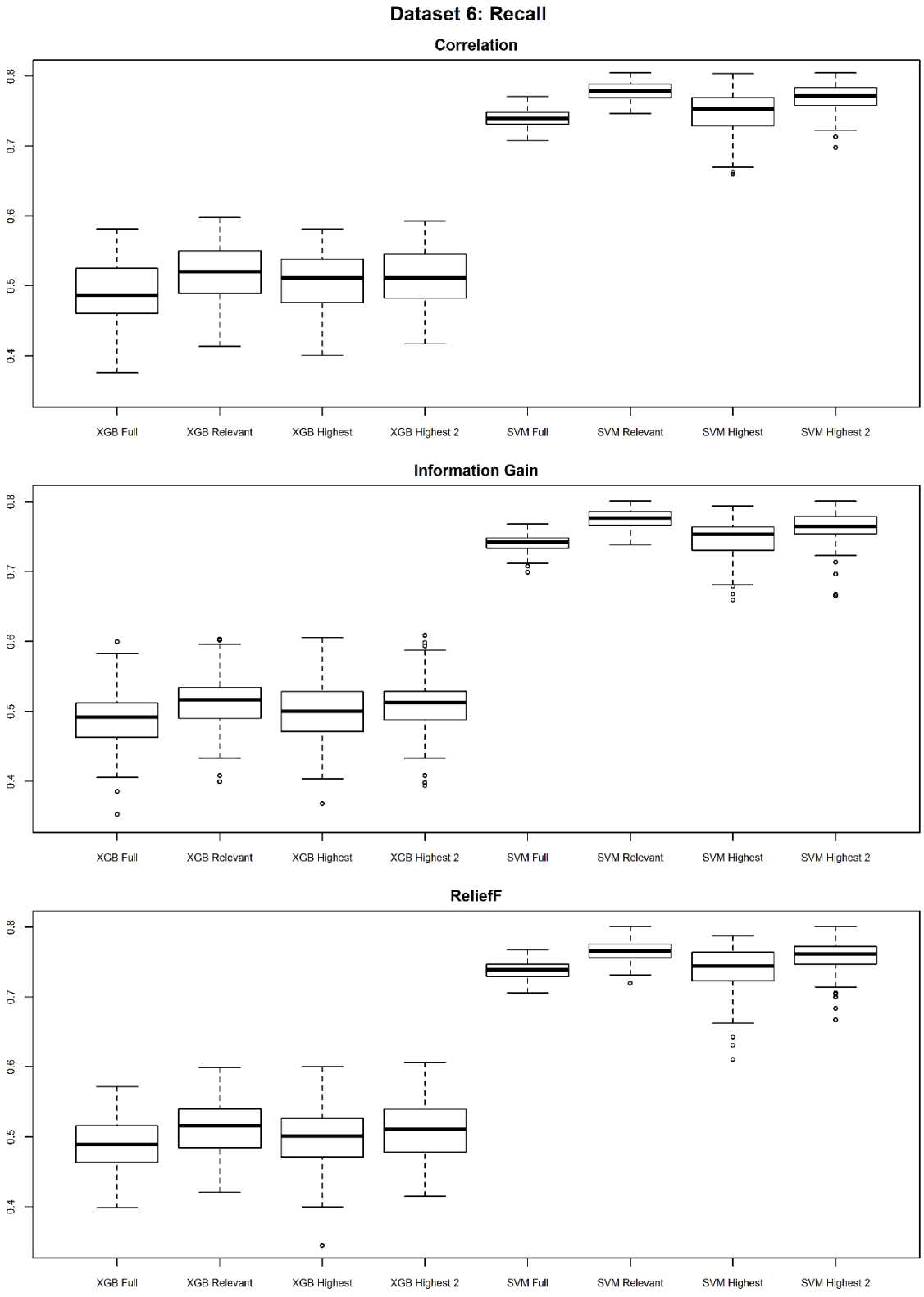


Figure F.24 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 6.

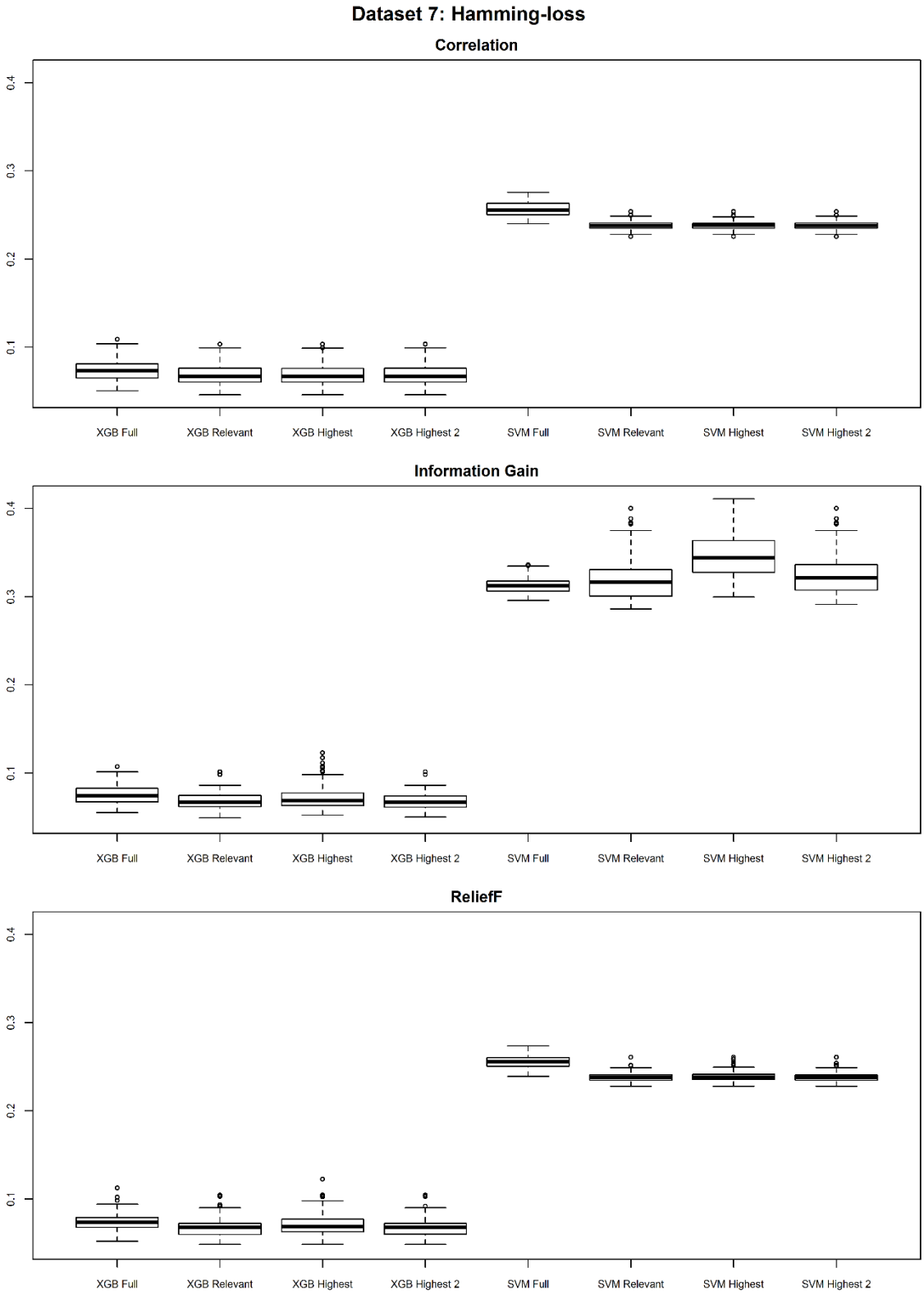


Figure F.25 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 7.

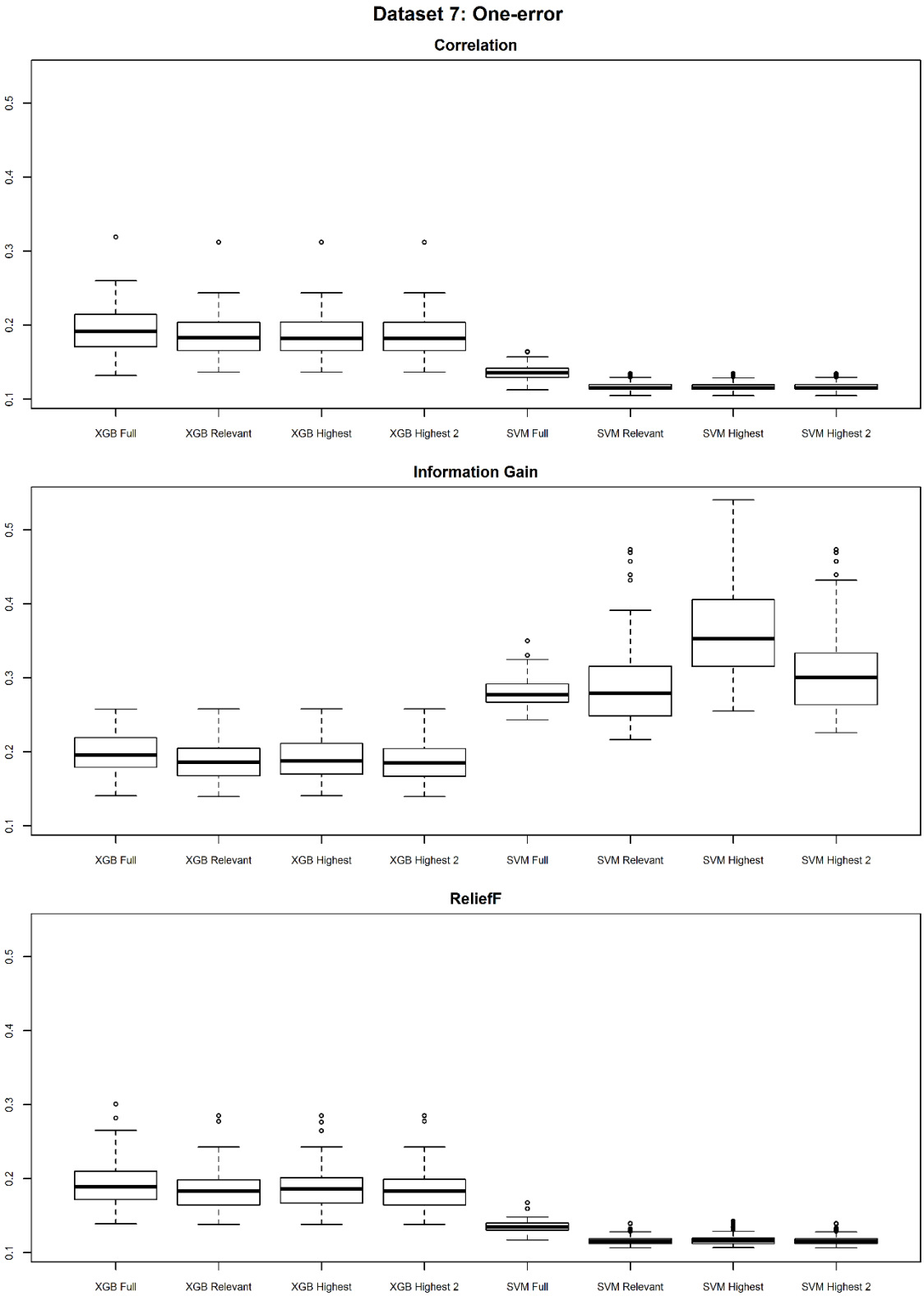


Figure F.26 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 7.

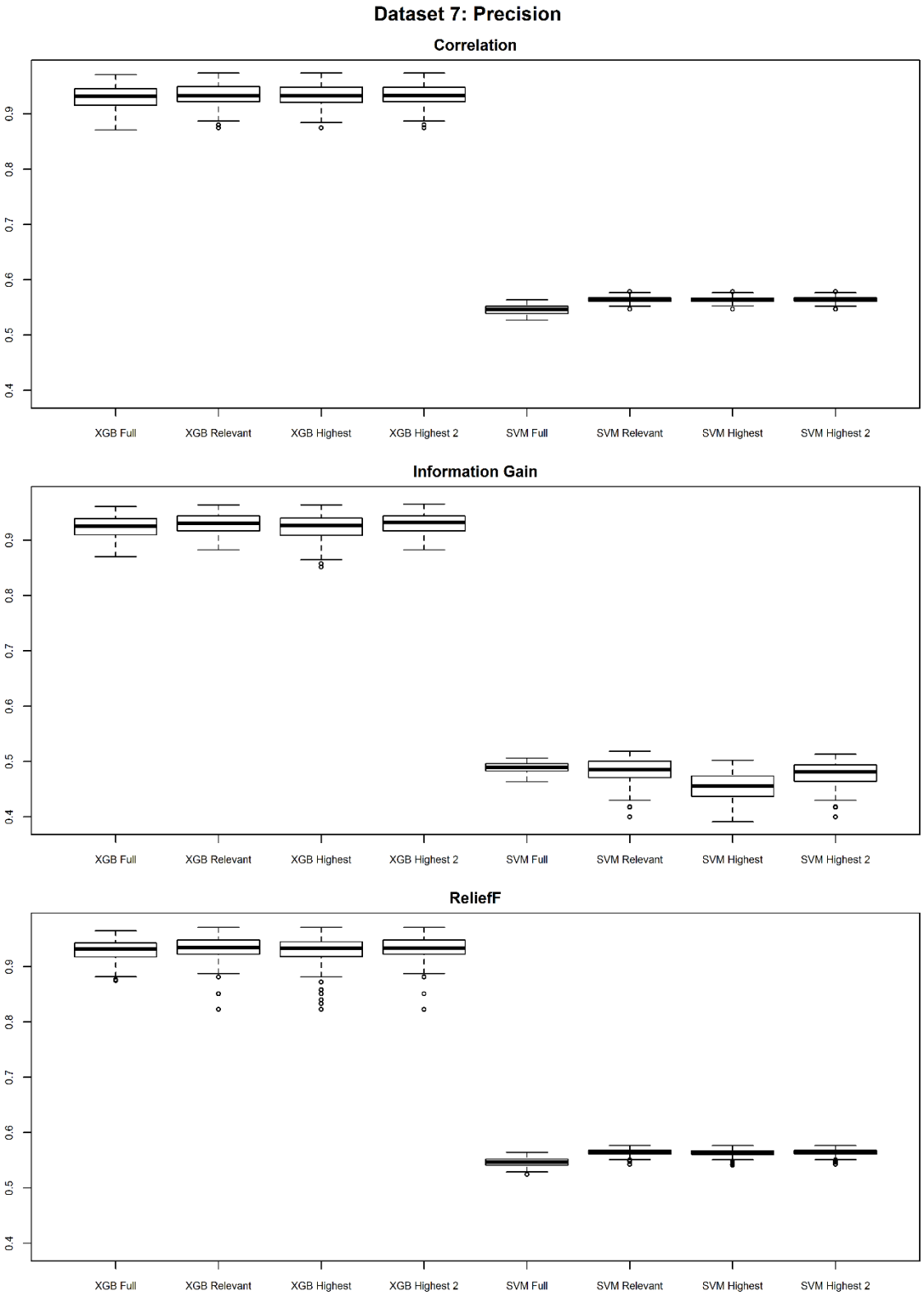


Figure F.27 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 7.

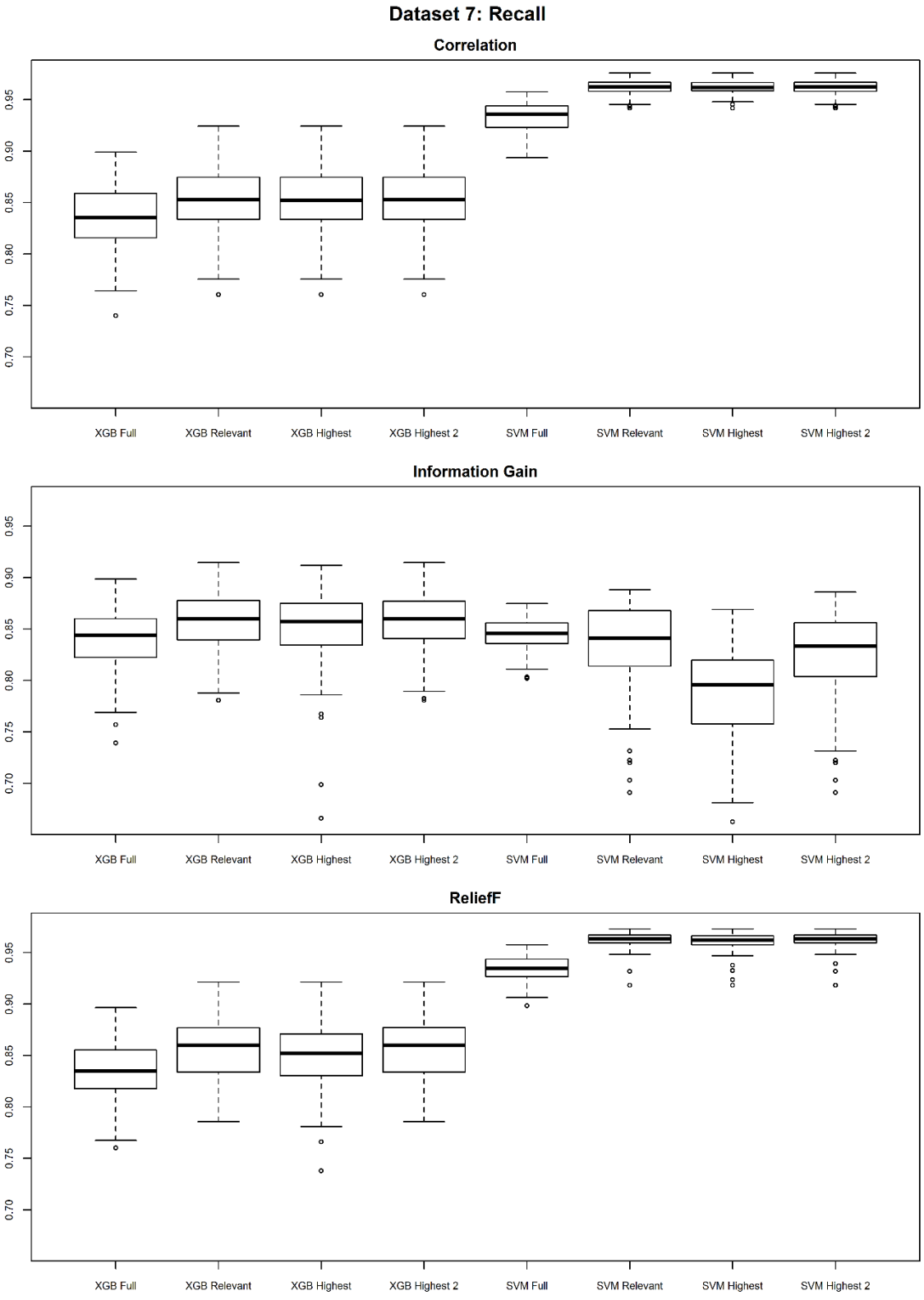


Figure F.28 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 7.

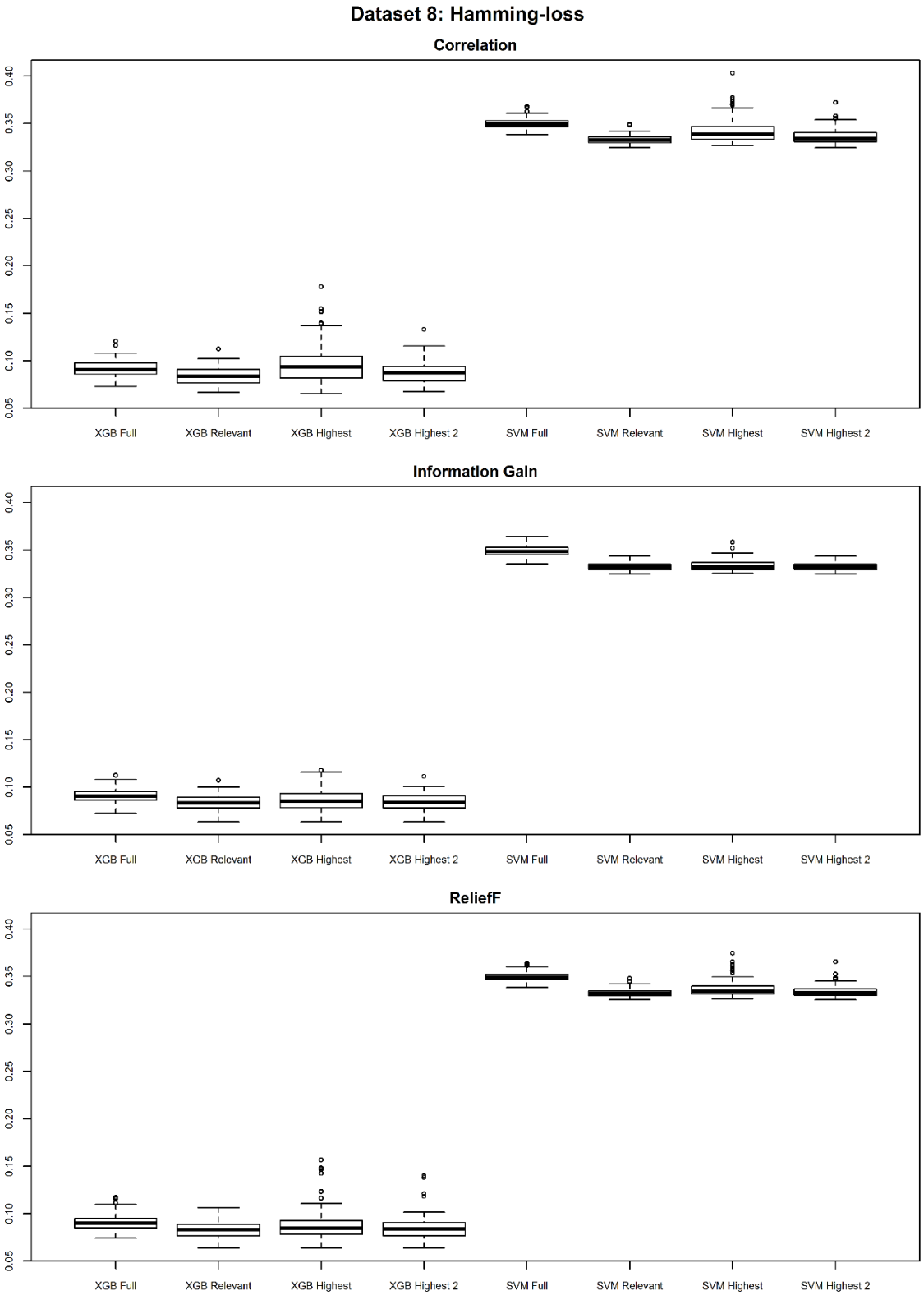


Figure F.29 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 8.

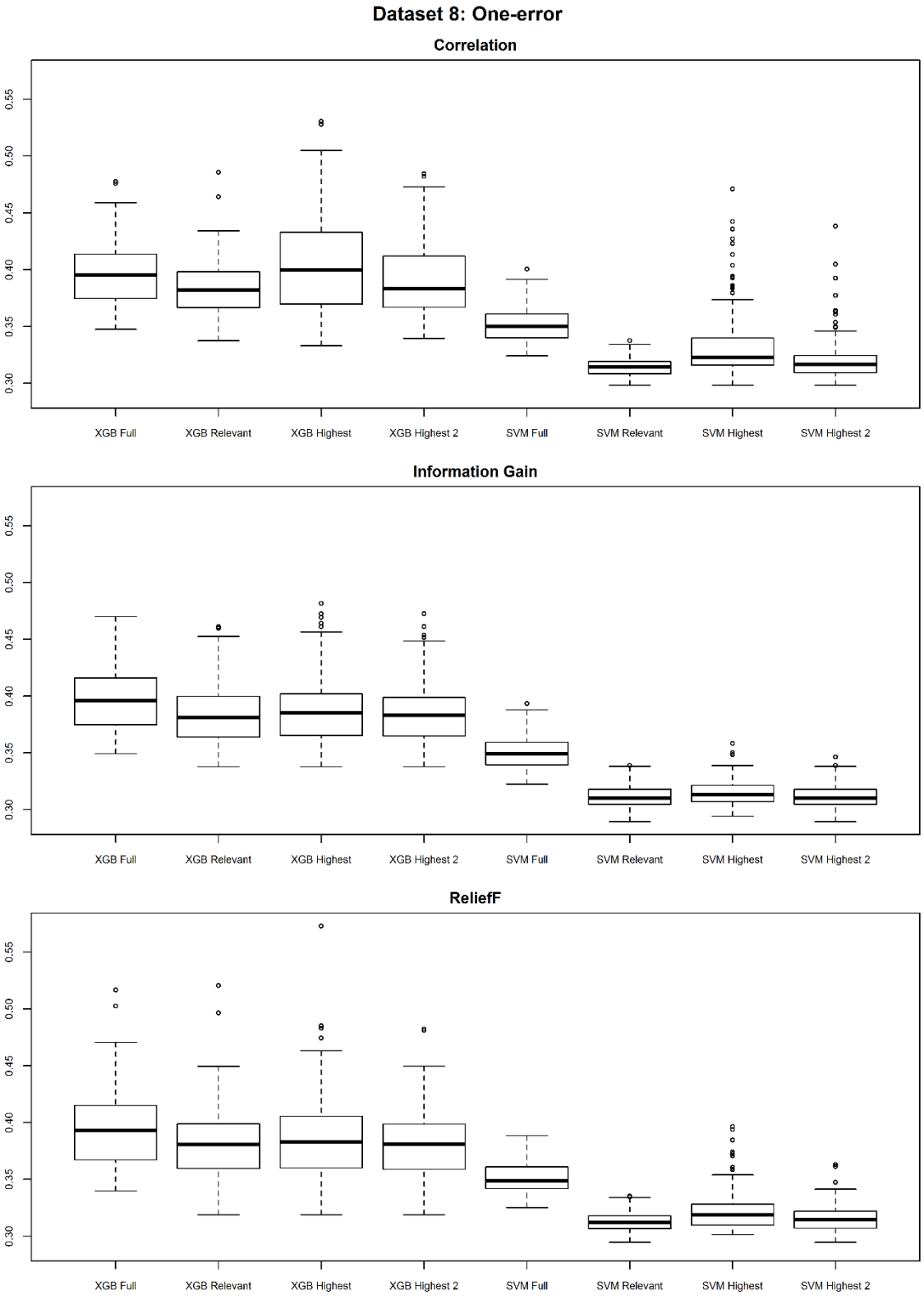


Figure F.30 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 8.

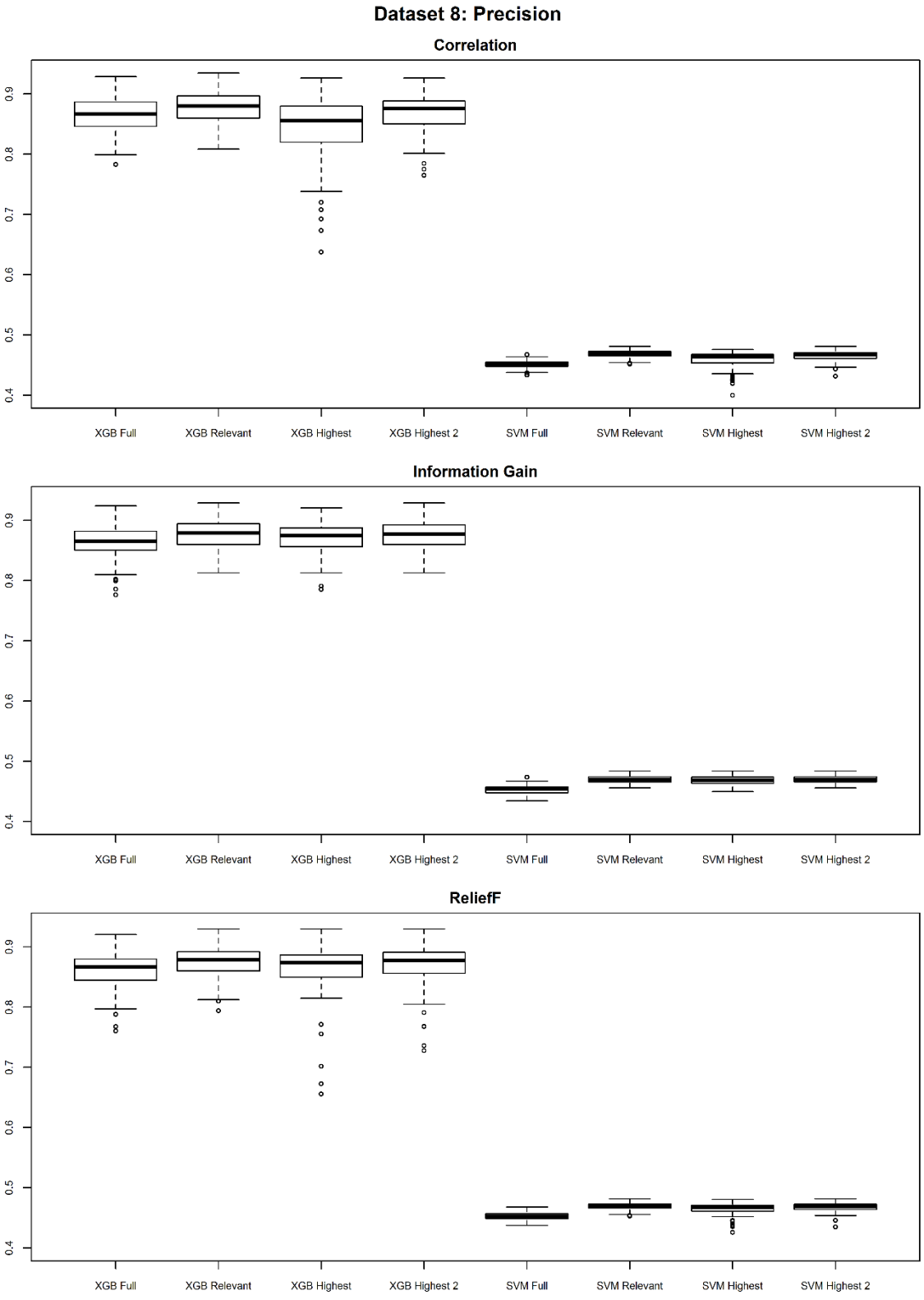


Figure F.31 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 8.

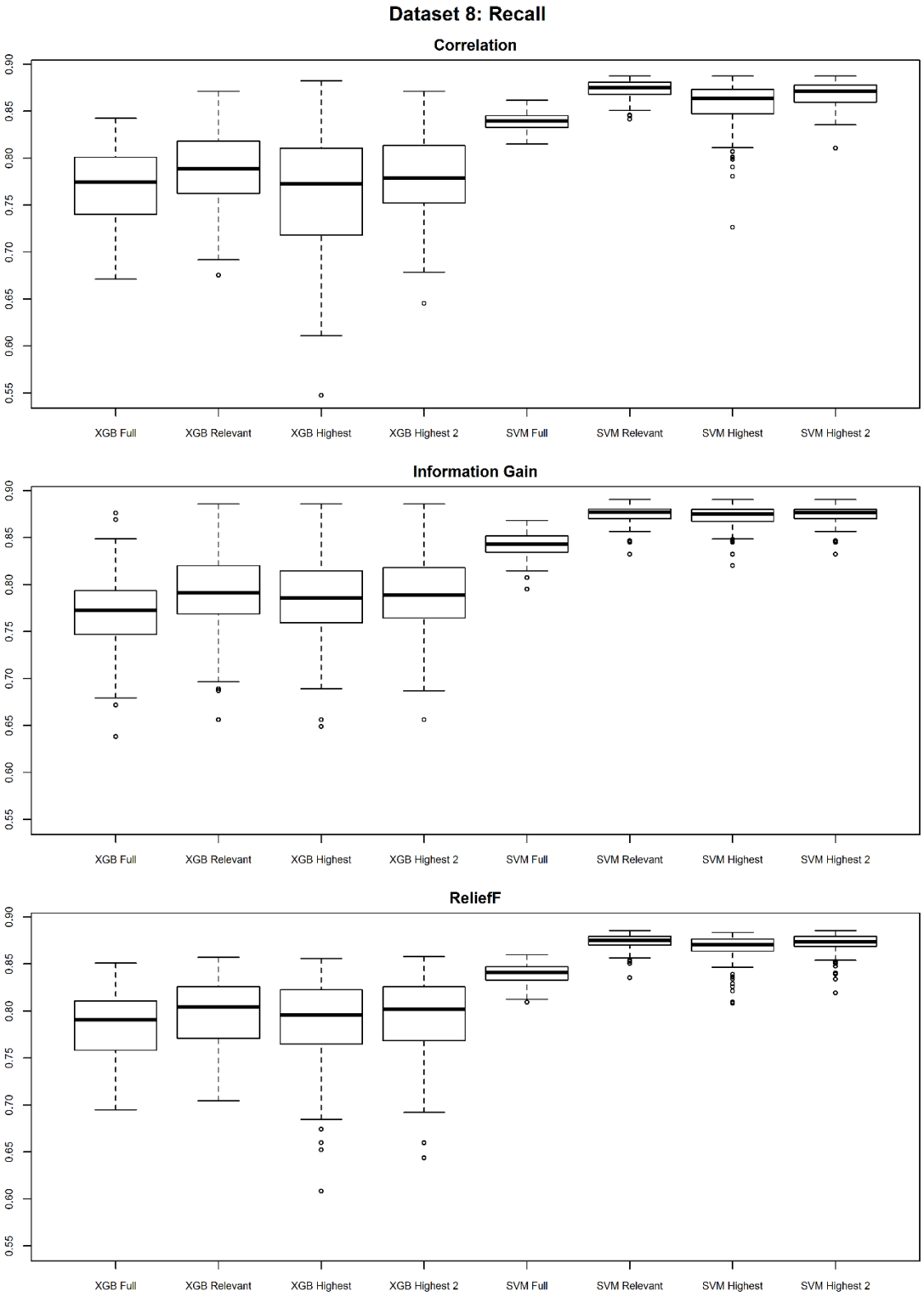


Figure F.32 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 8.

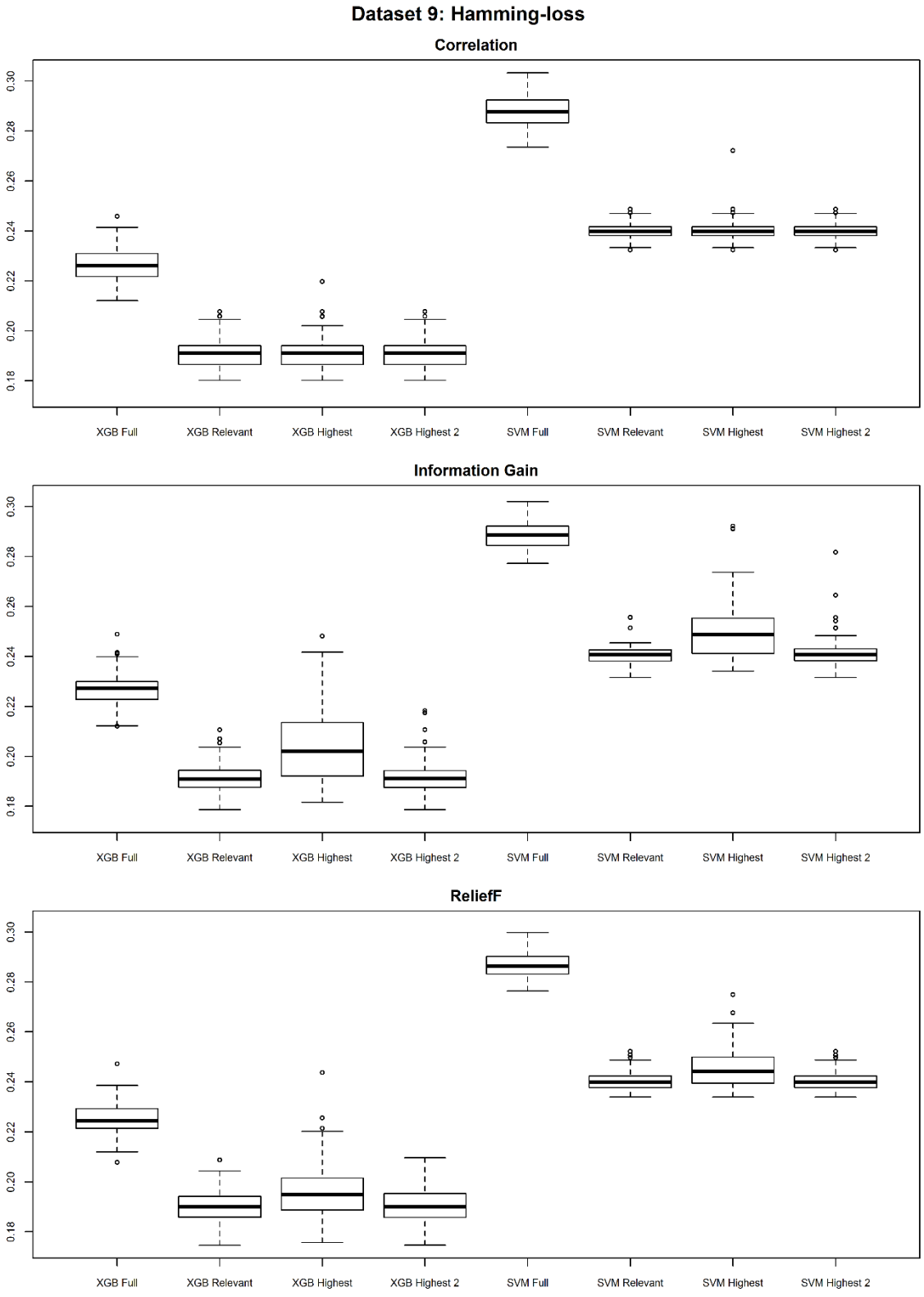


Figure F.33 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 9.

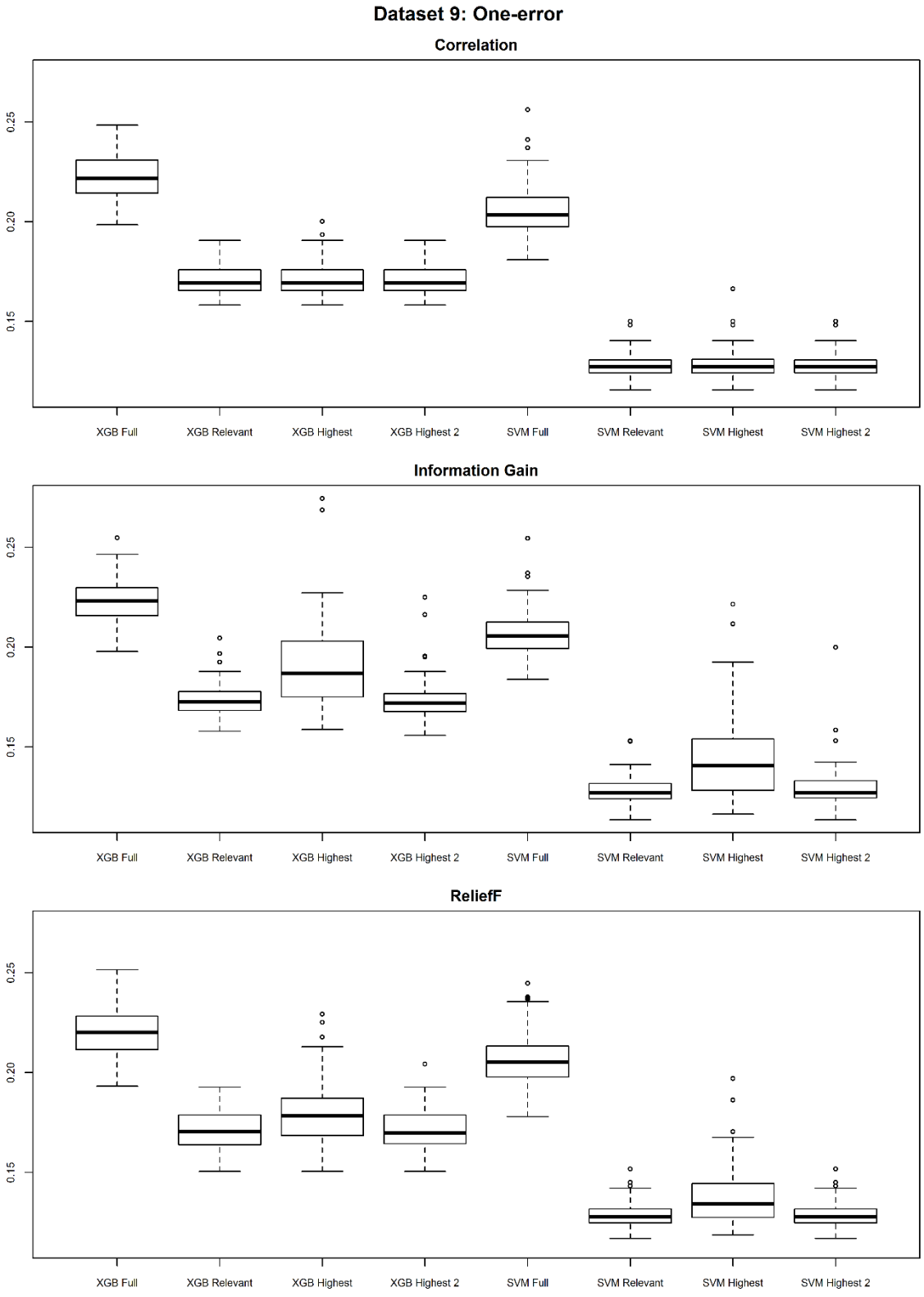


Figure F.34 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 9.

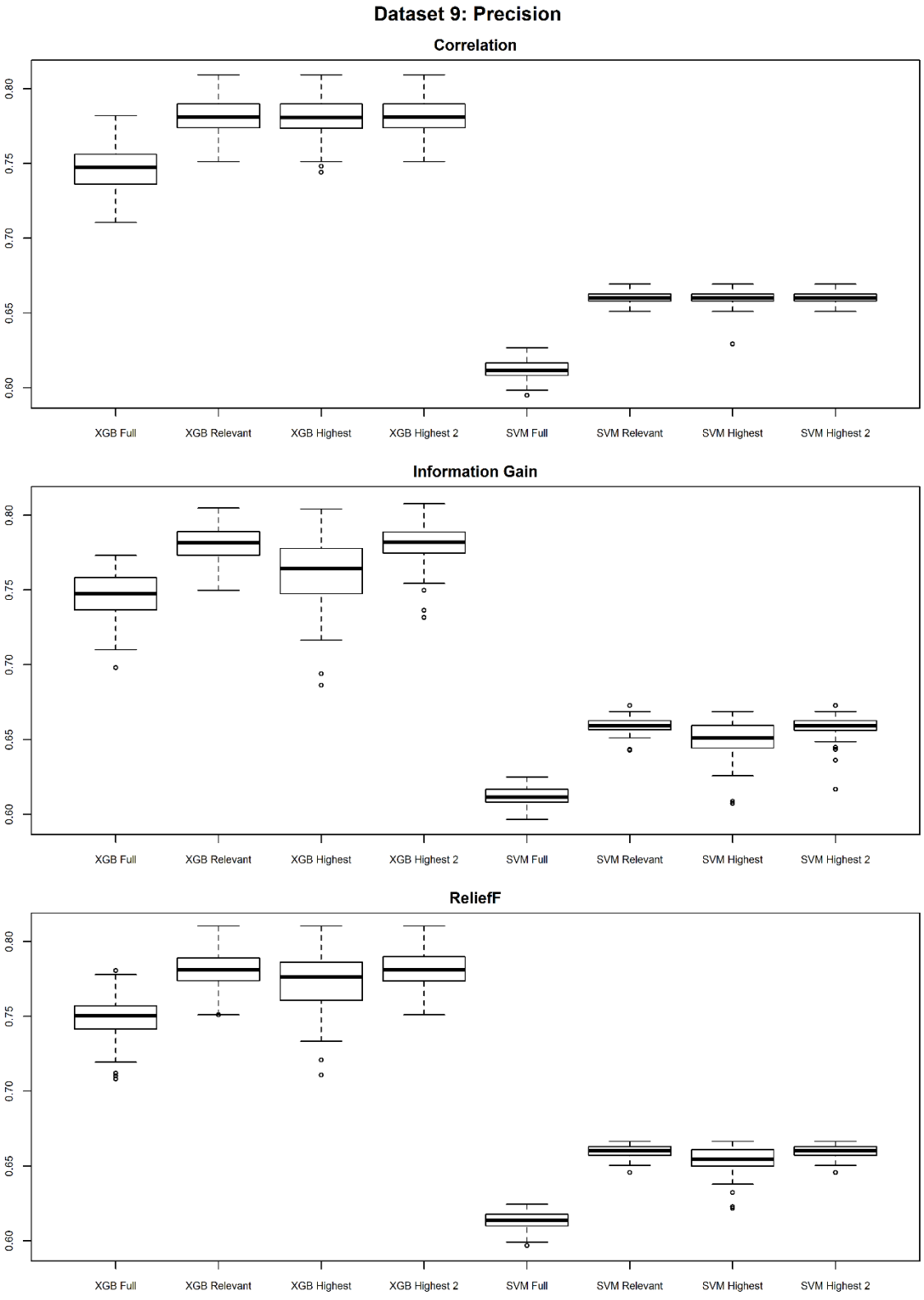


Figure F.35 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 9.

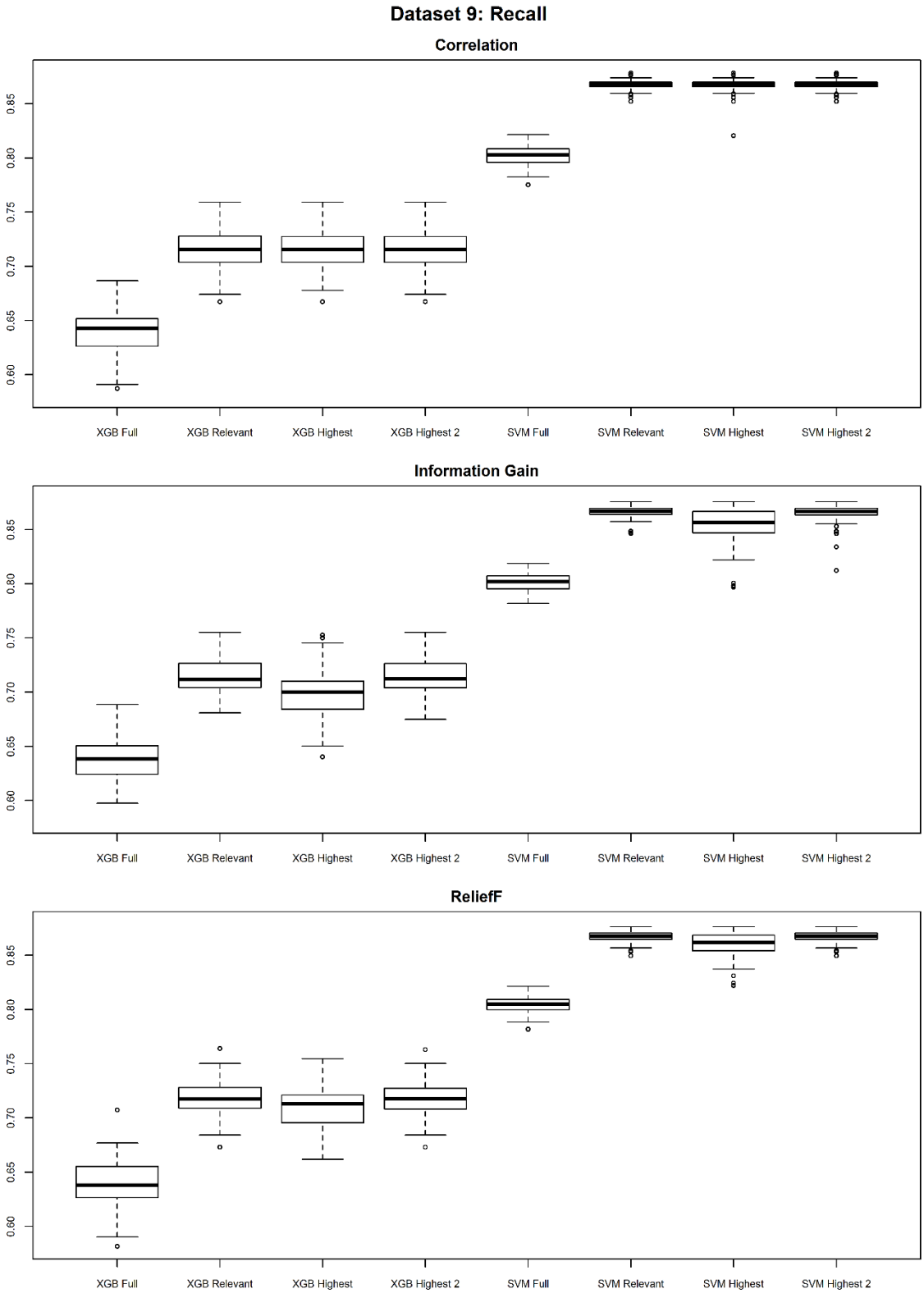


Figure F.36 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 9.

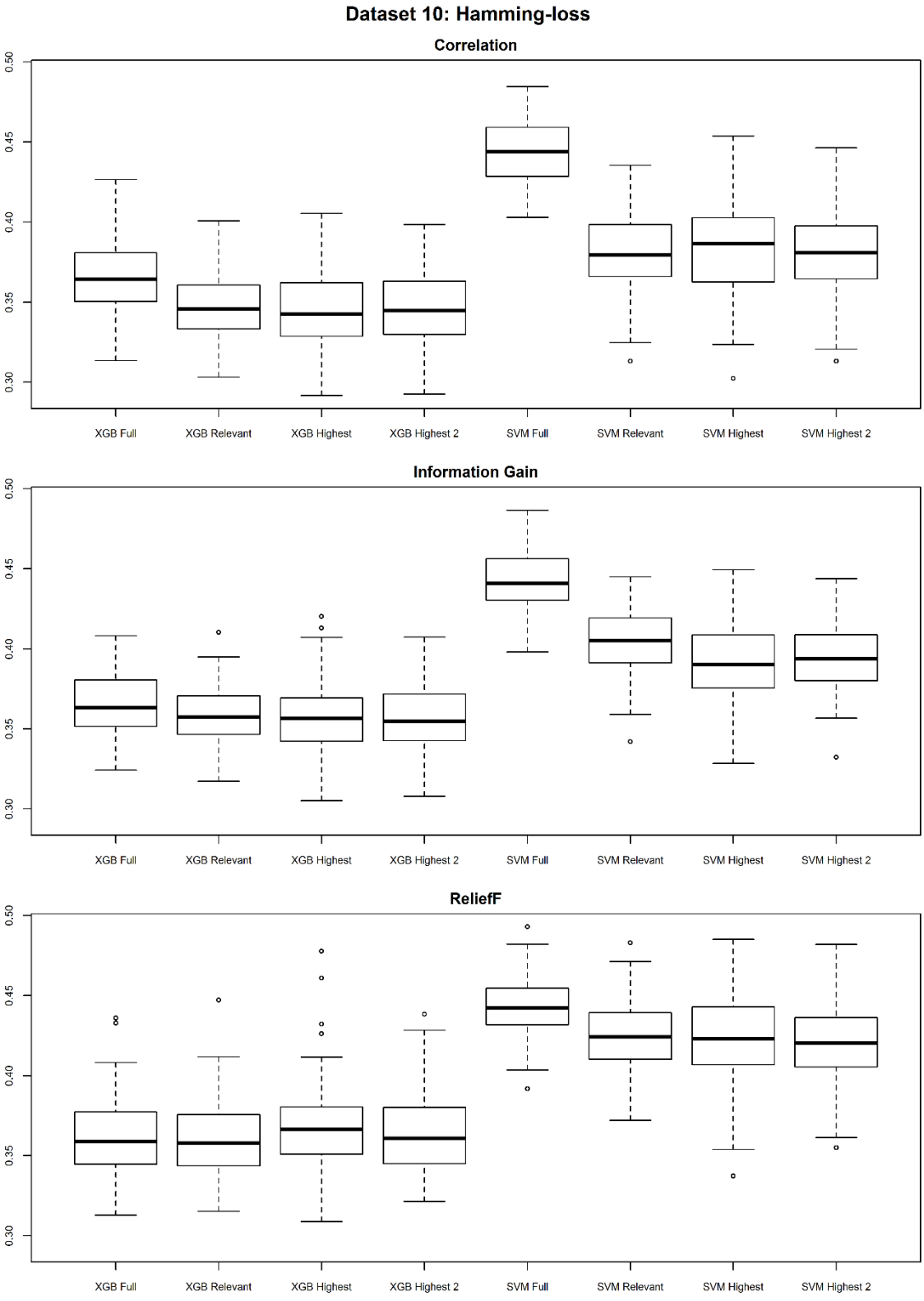


Figure F.37 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 10.

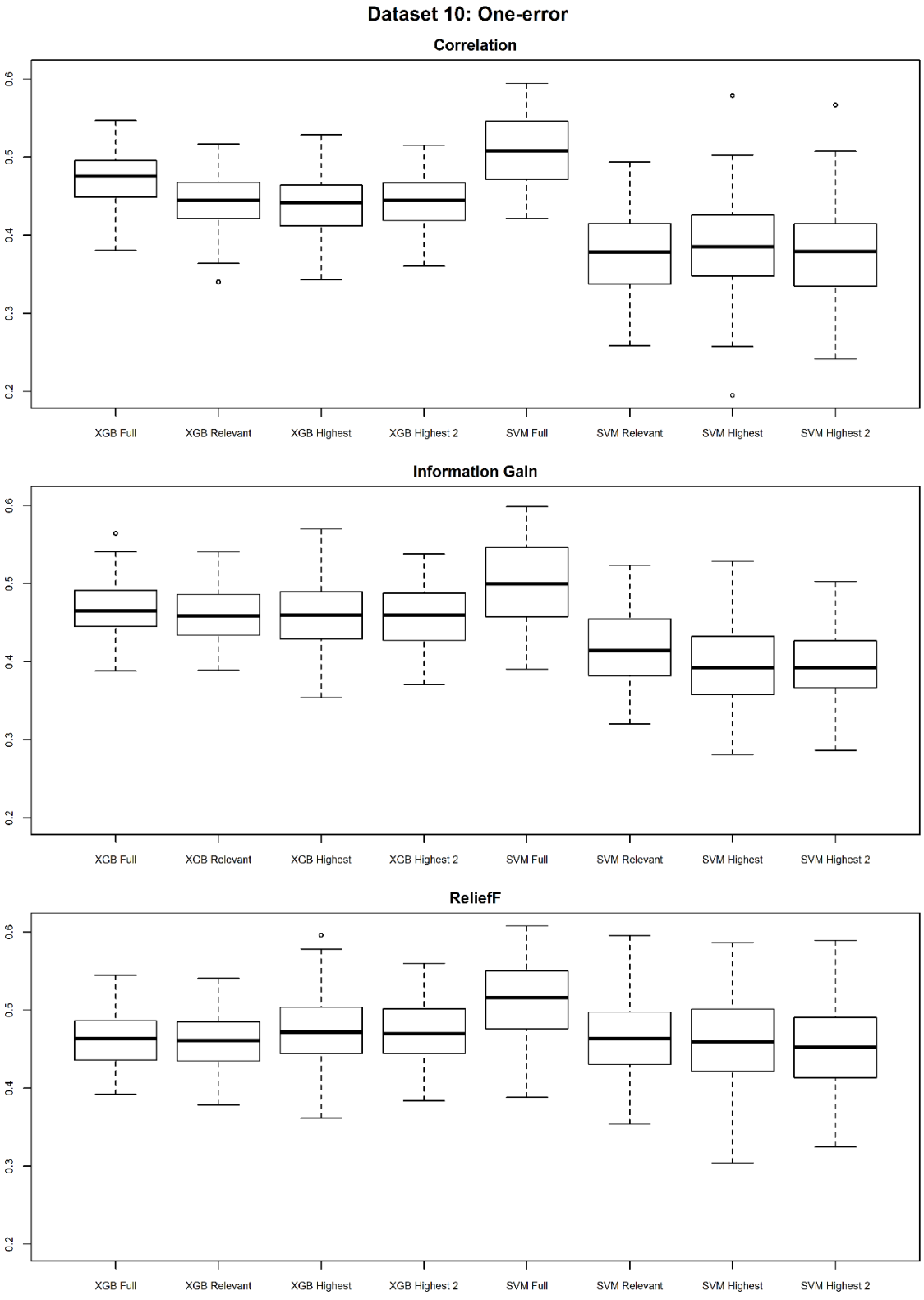


Figure F.38 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 10.

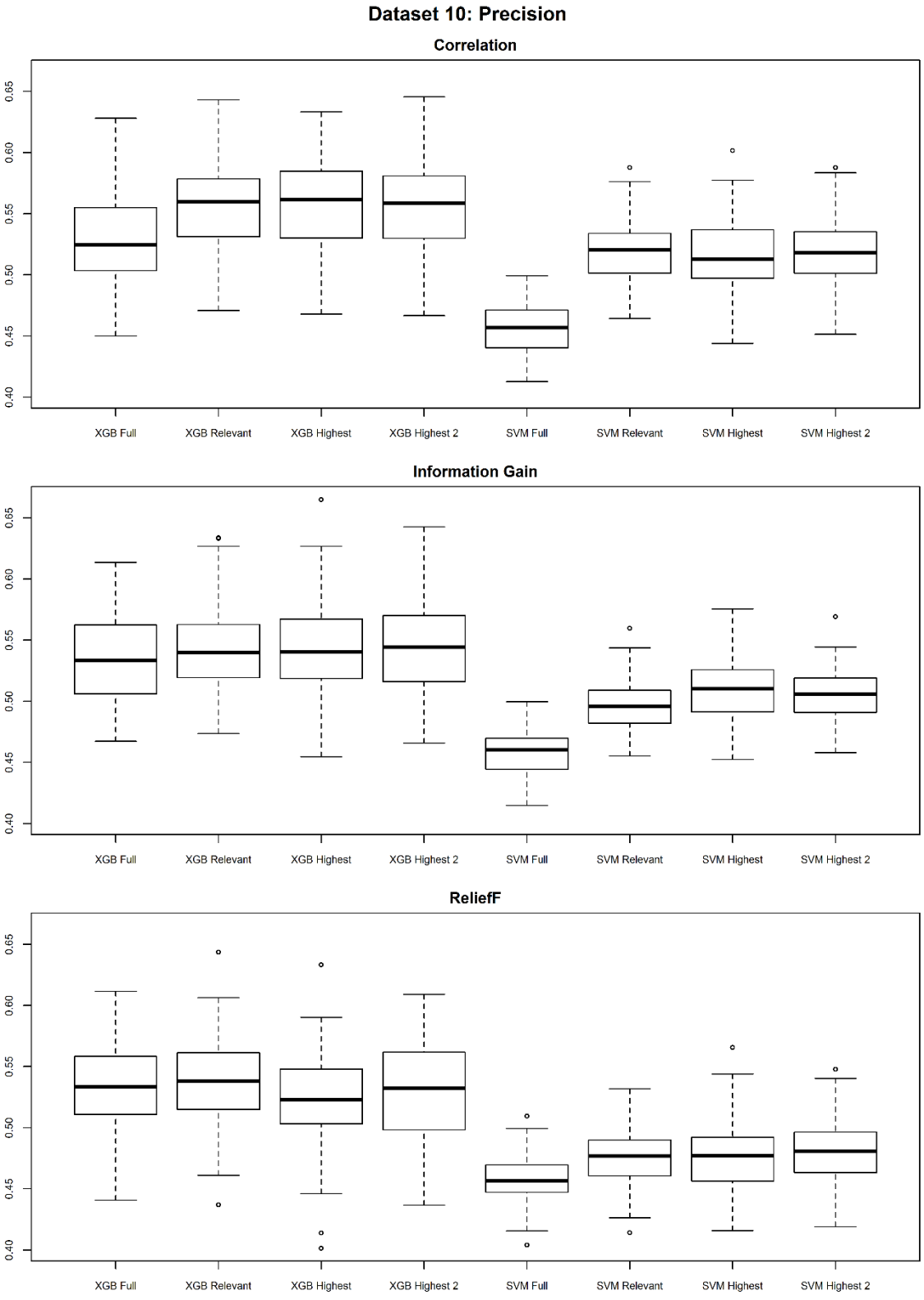


Figure F.39 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 10.

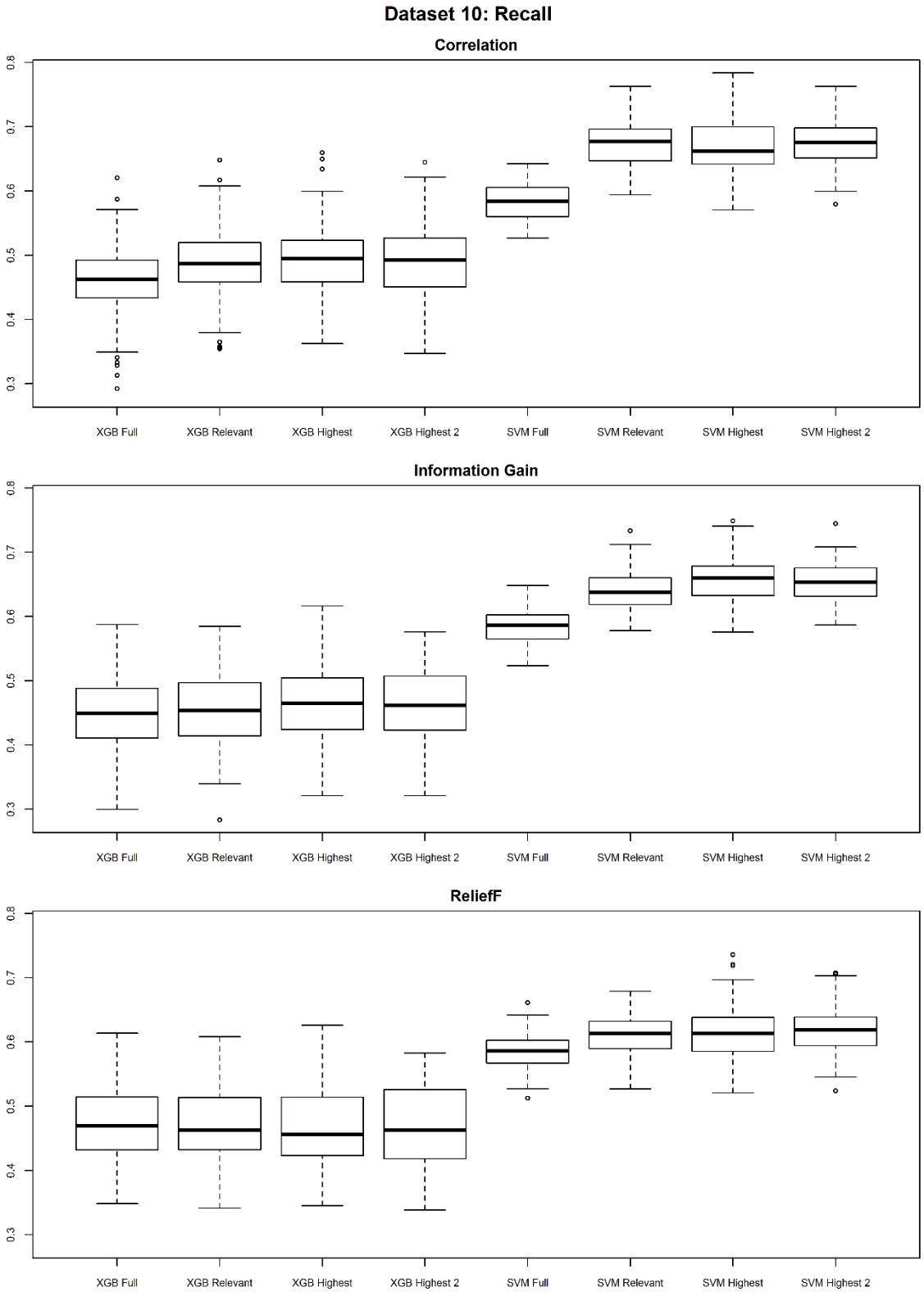


Figure F.40 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 10.

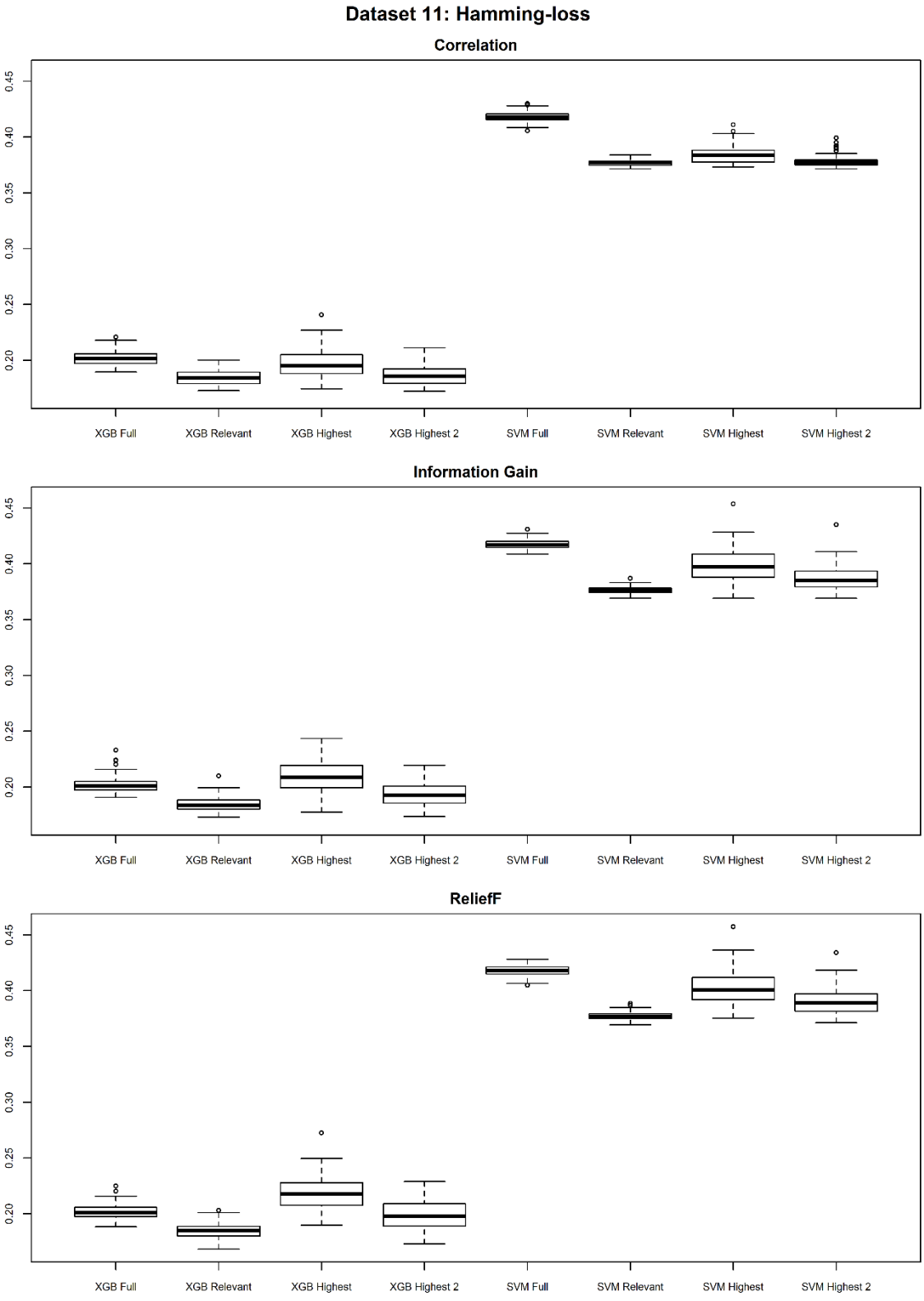


Figure F.41 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 11.

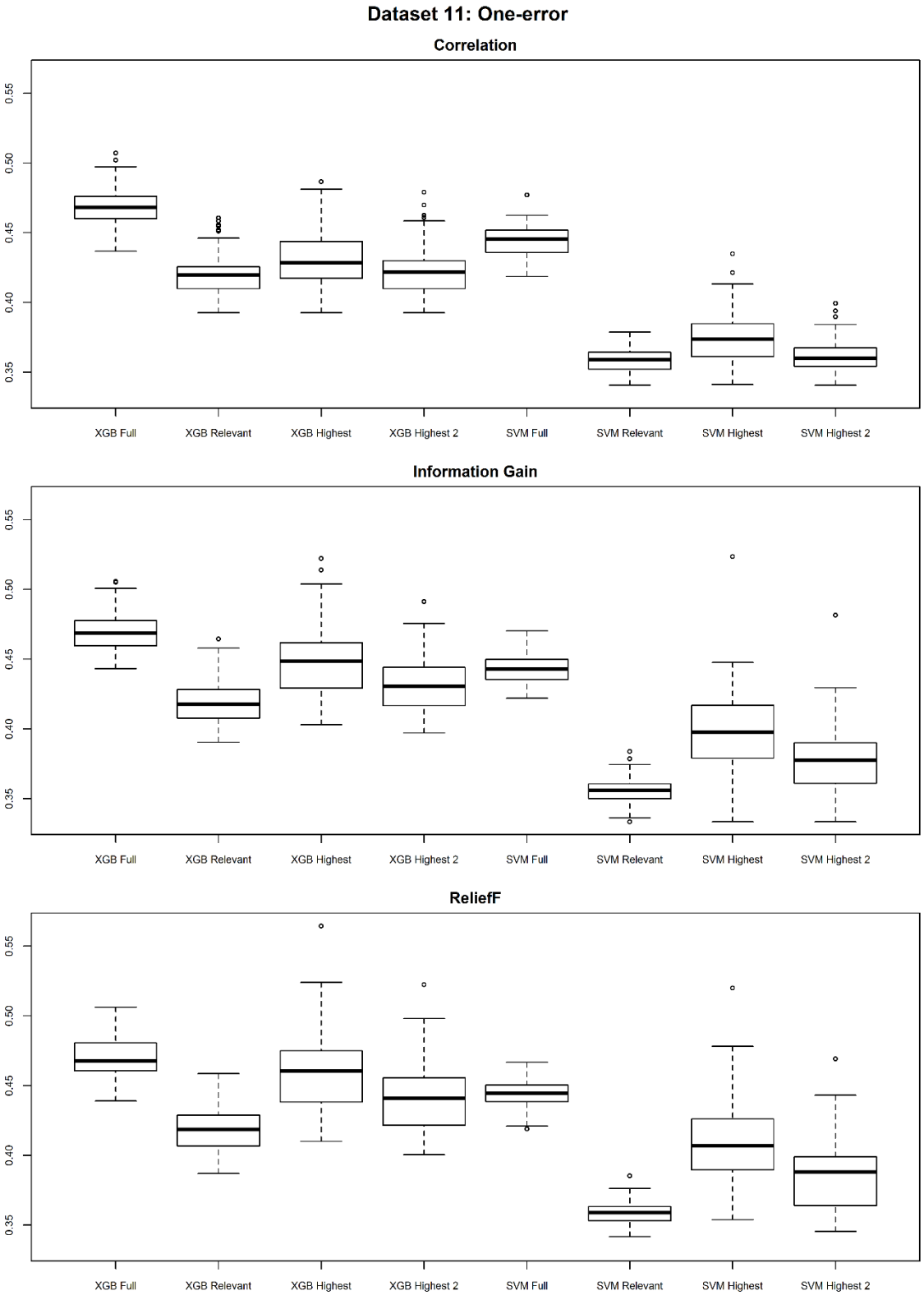


Figure F.42 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 11.

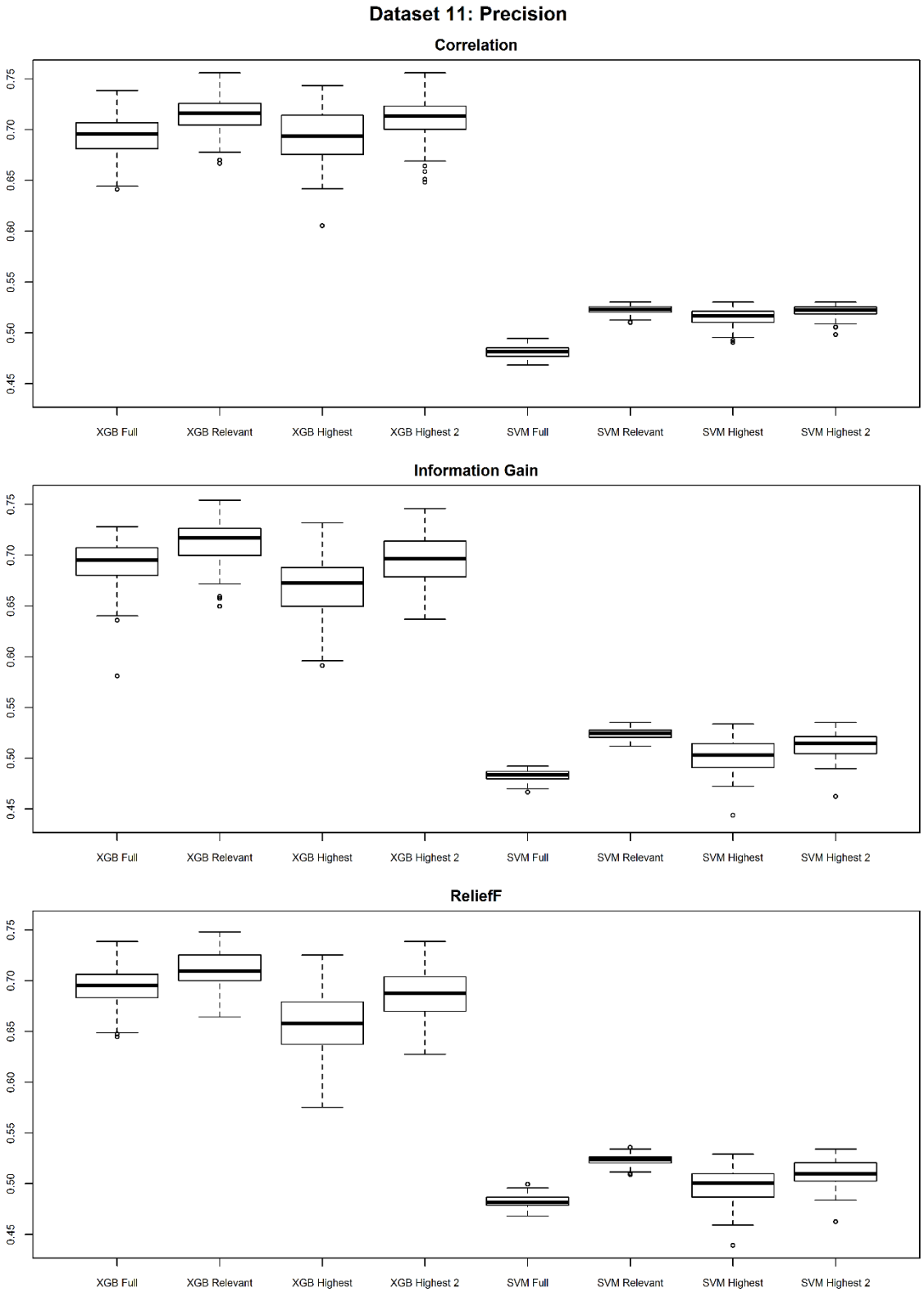


Figure F.43 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 11.

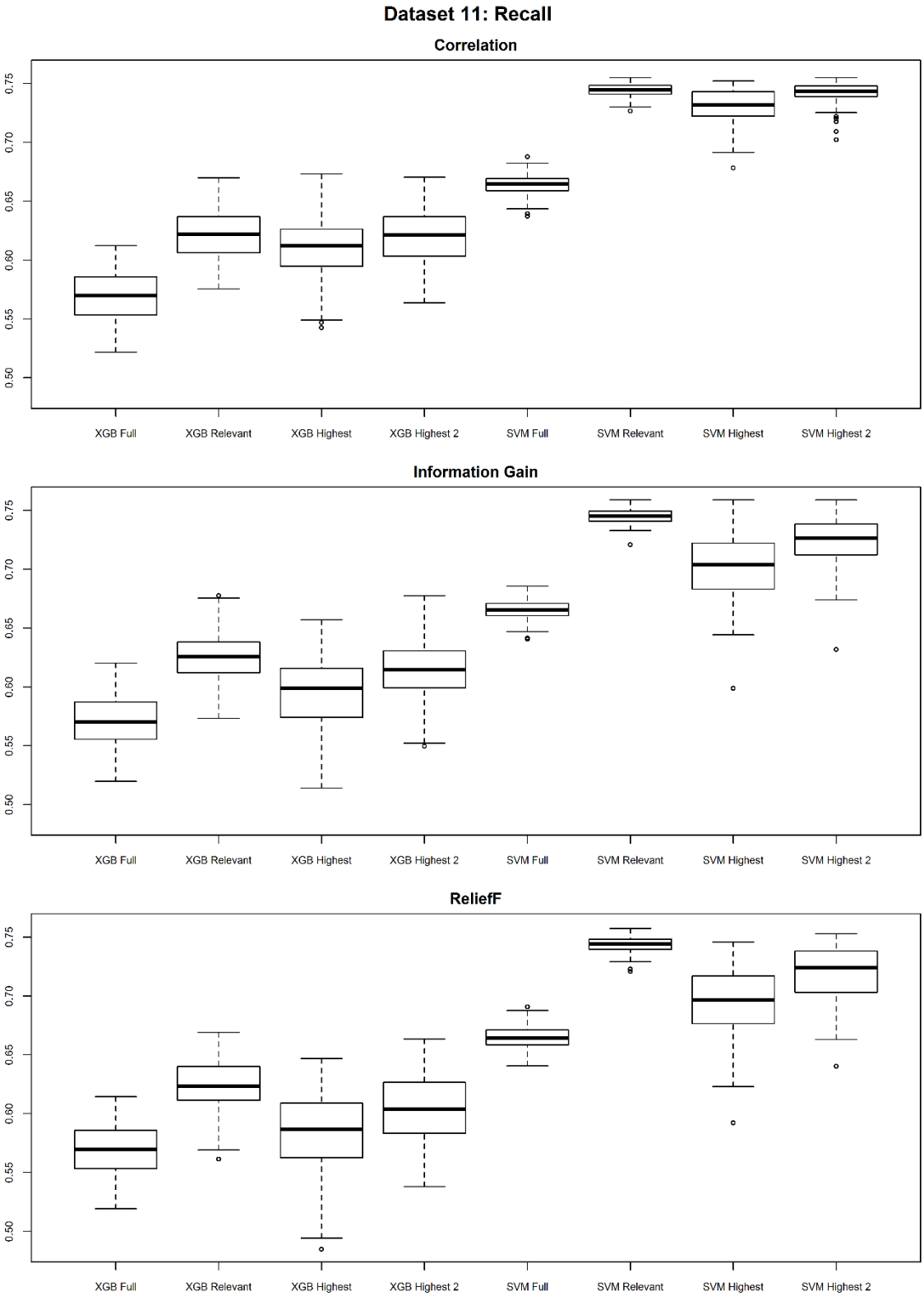


Figure F.44 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 11.

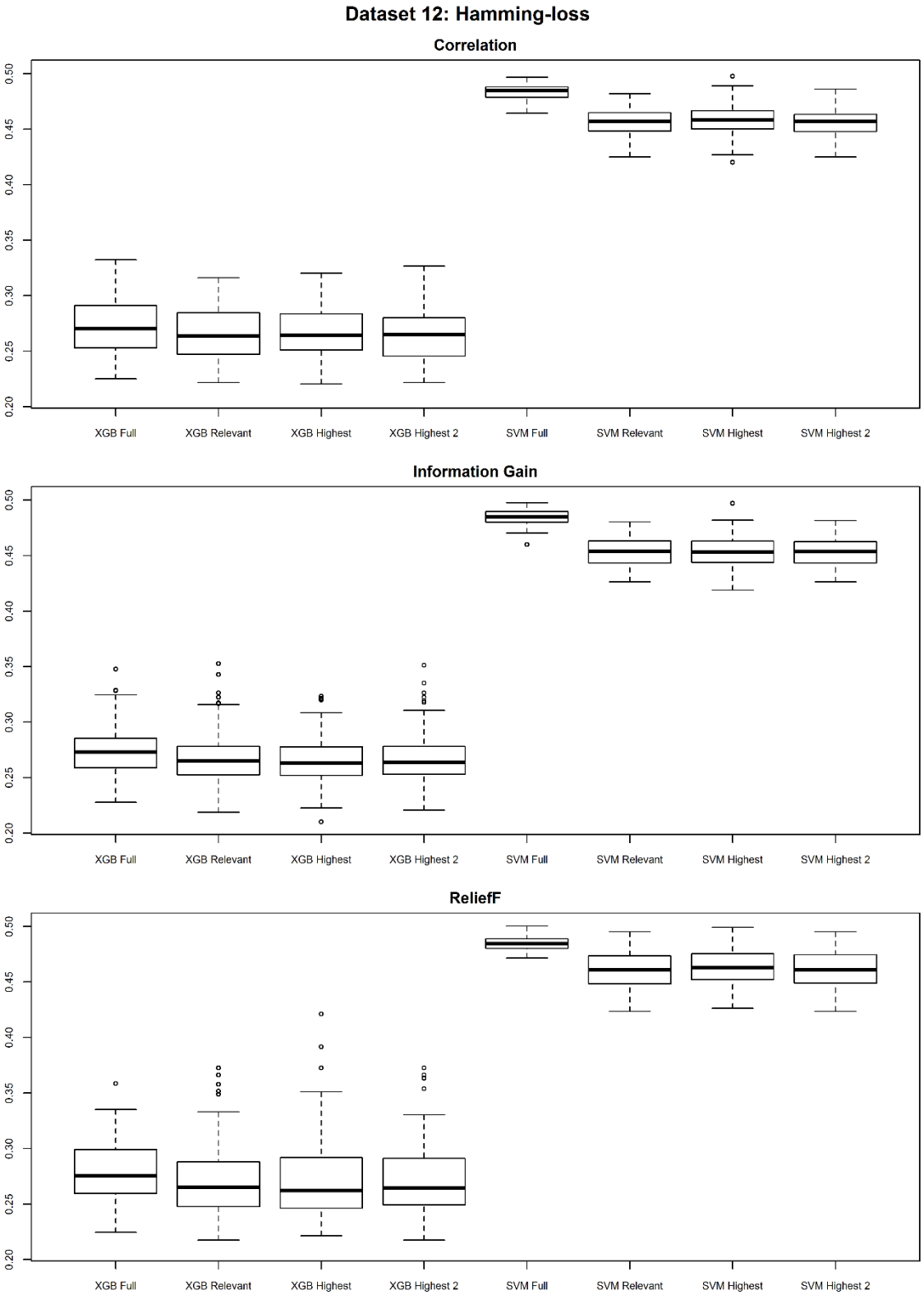


Figure F.45 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 12.

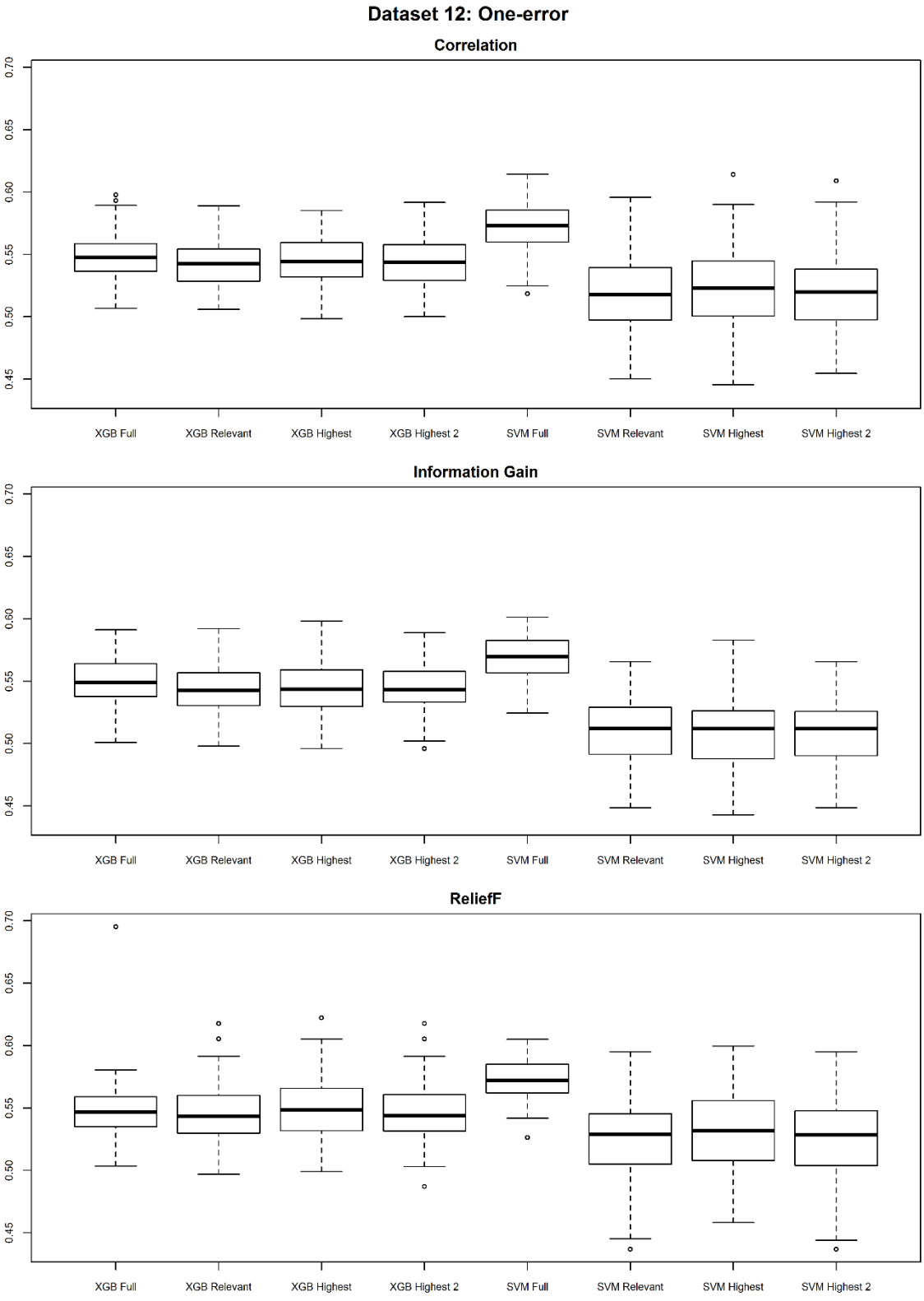


Figure F.46 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 12.

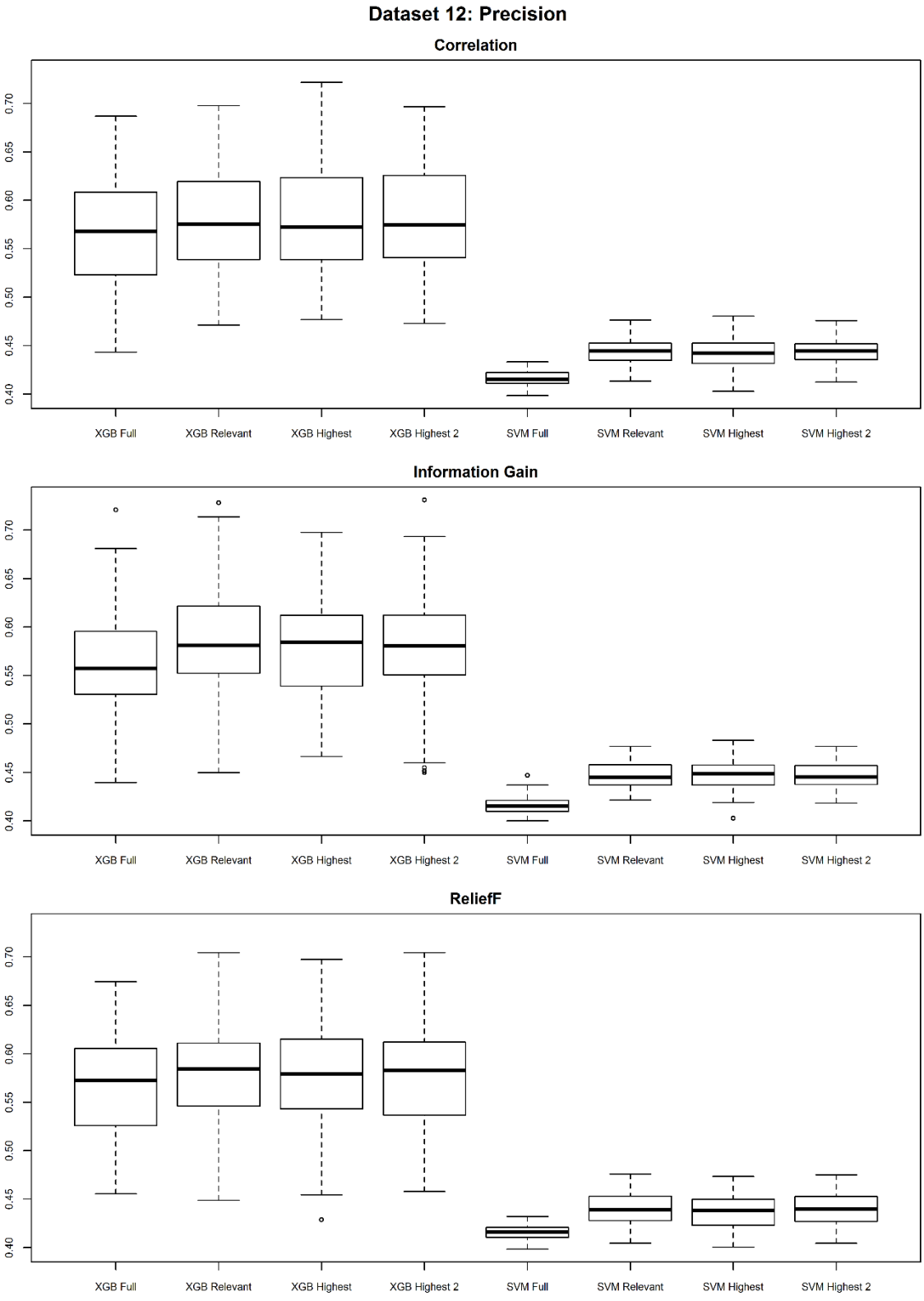


Figure F.47 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 12.

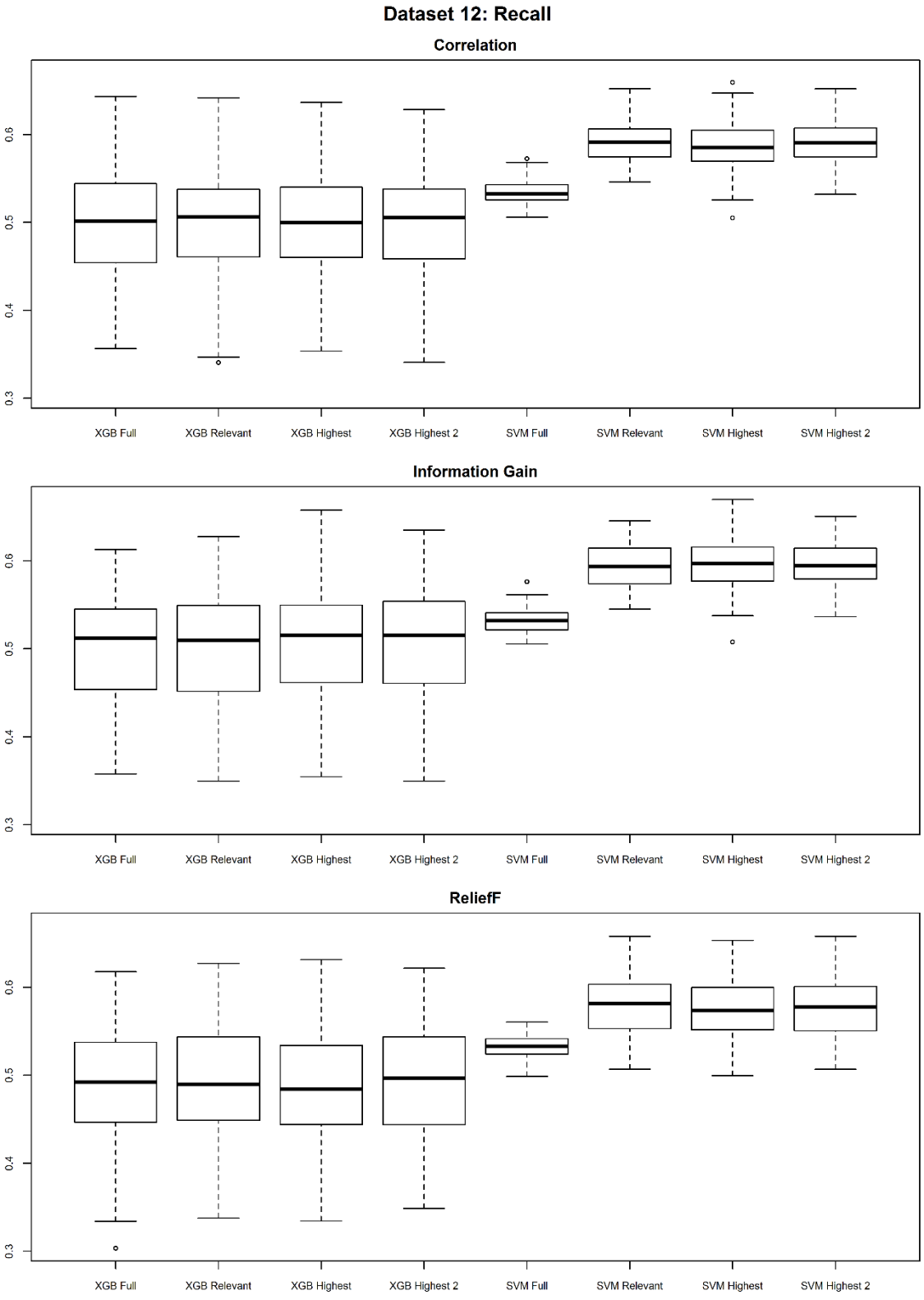


Figure F.48 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 12.

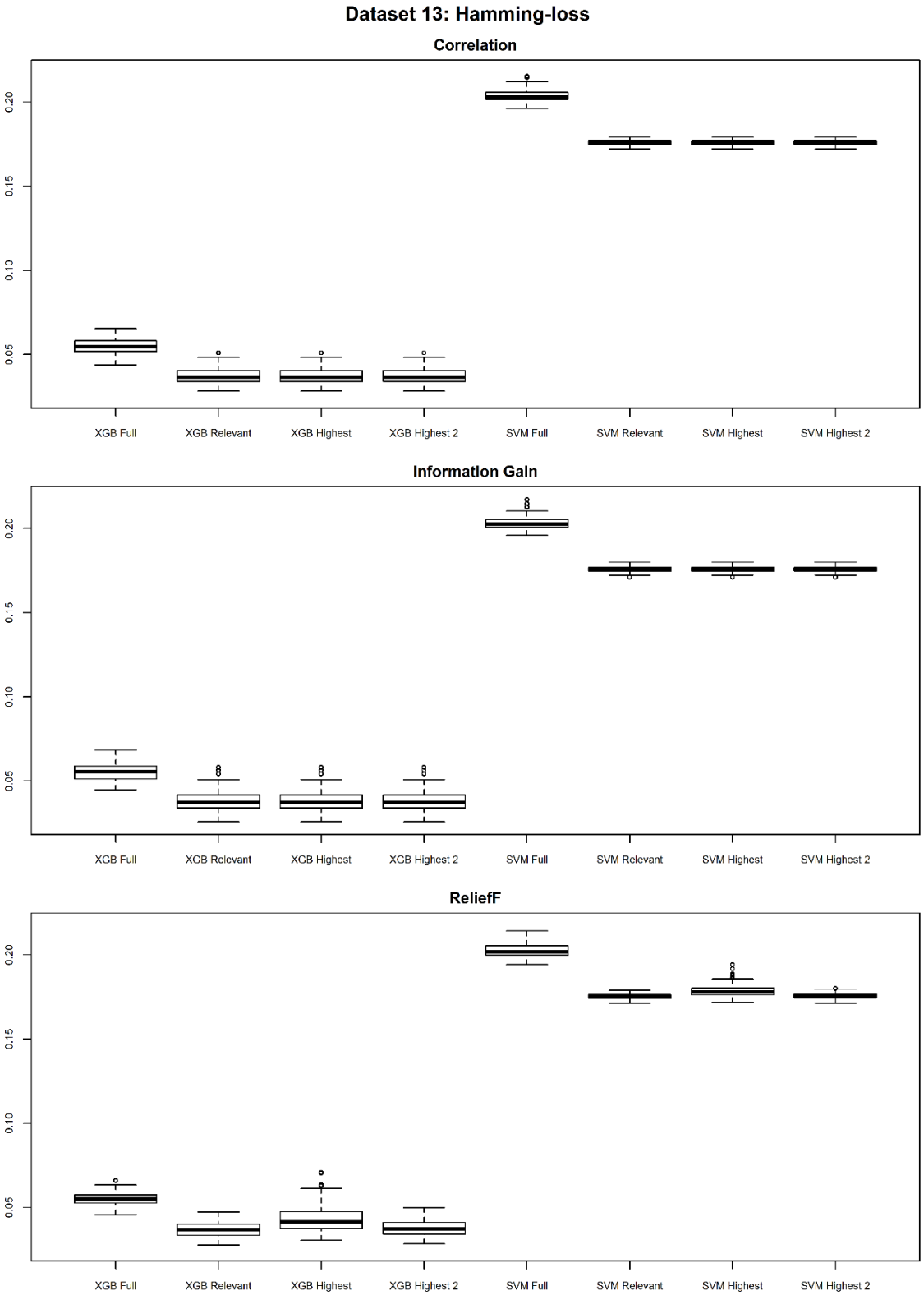


Figure F.49 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 13.

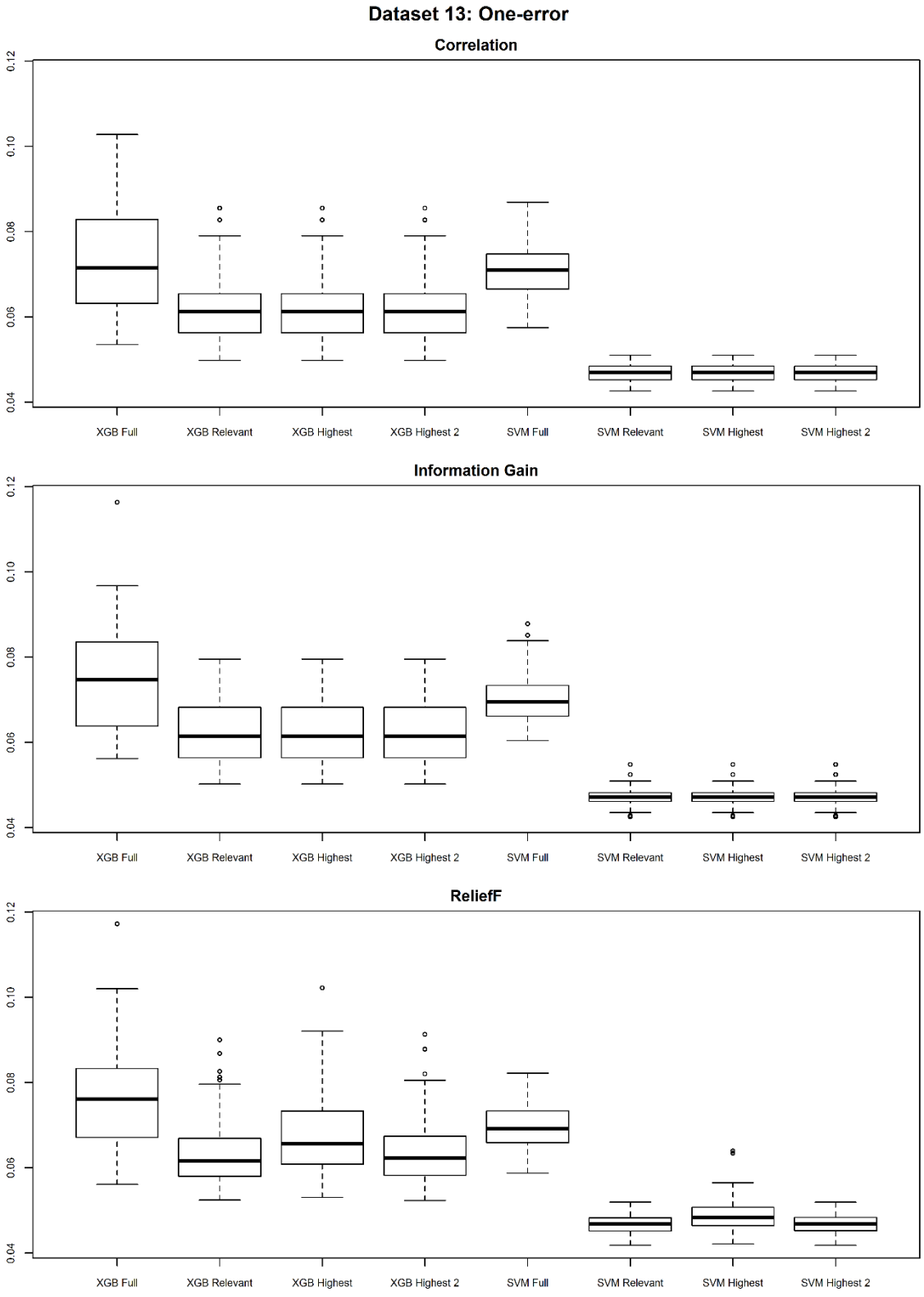


Figure F.50 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 13.

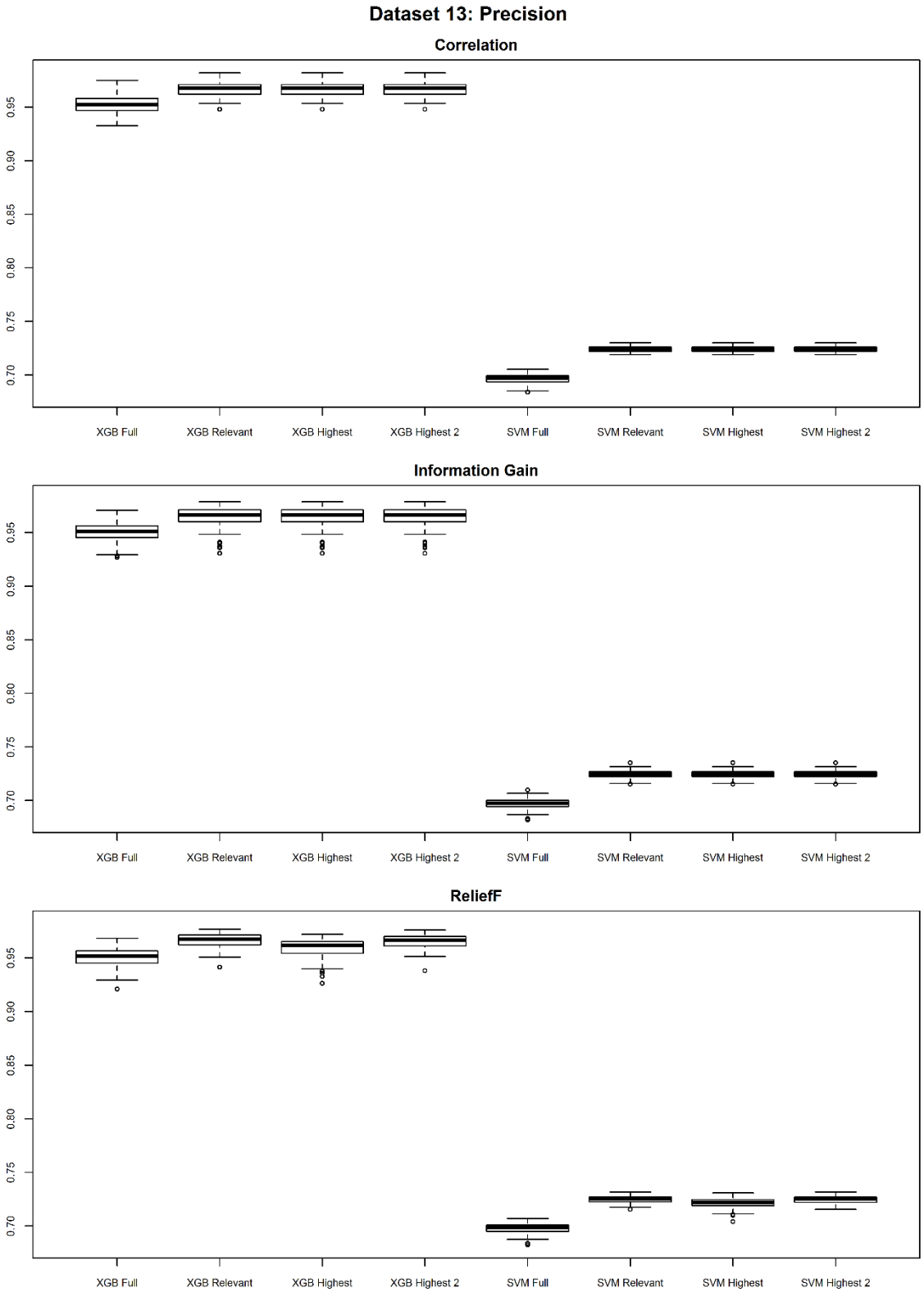


Figure F.51 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 13.

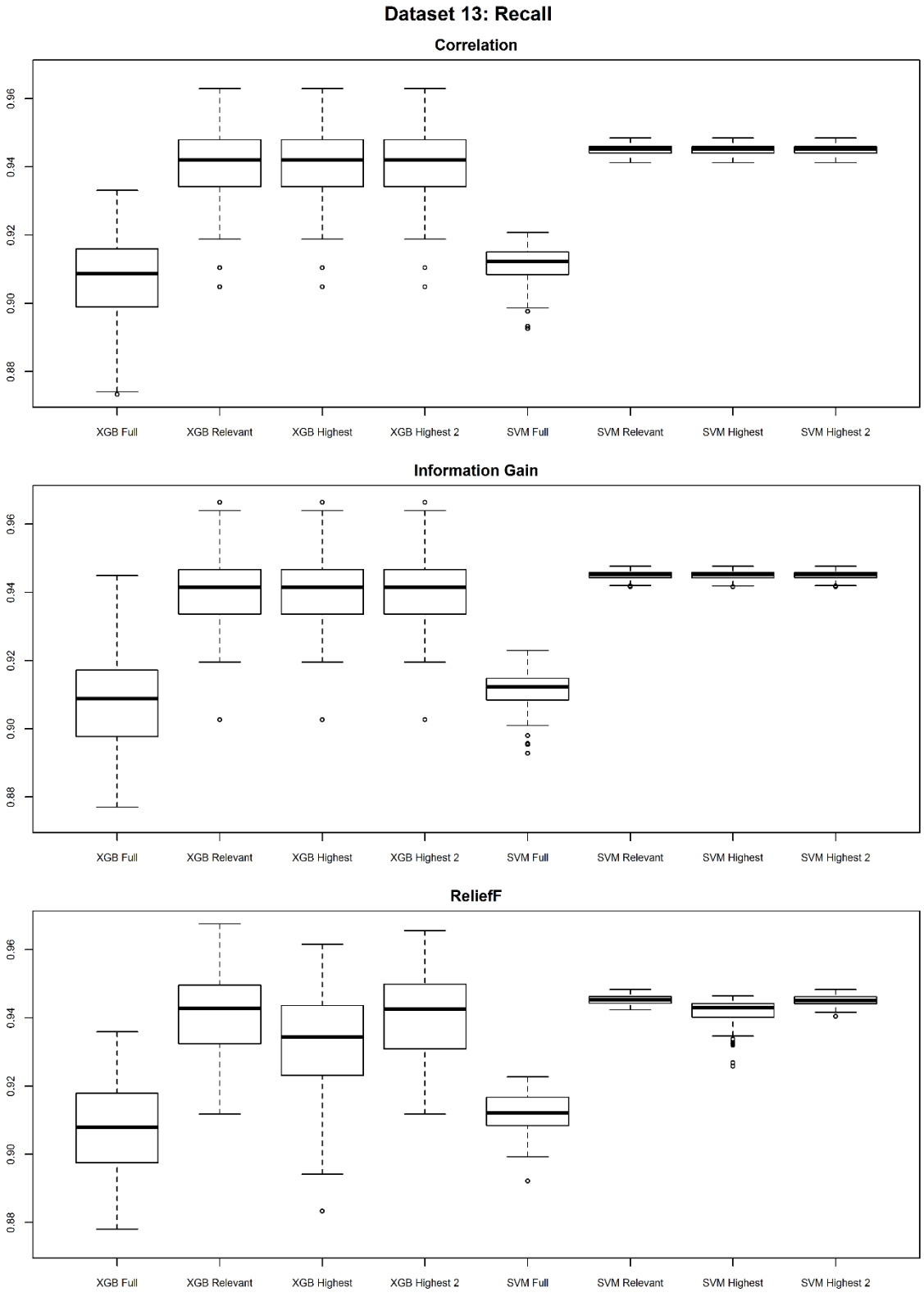


Figure F.52 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 13.

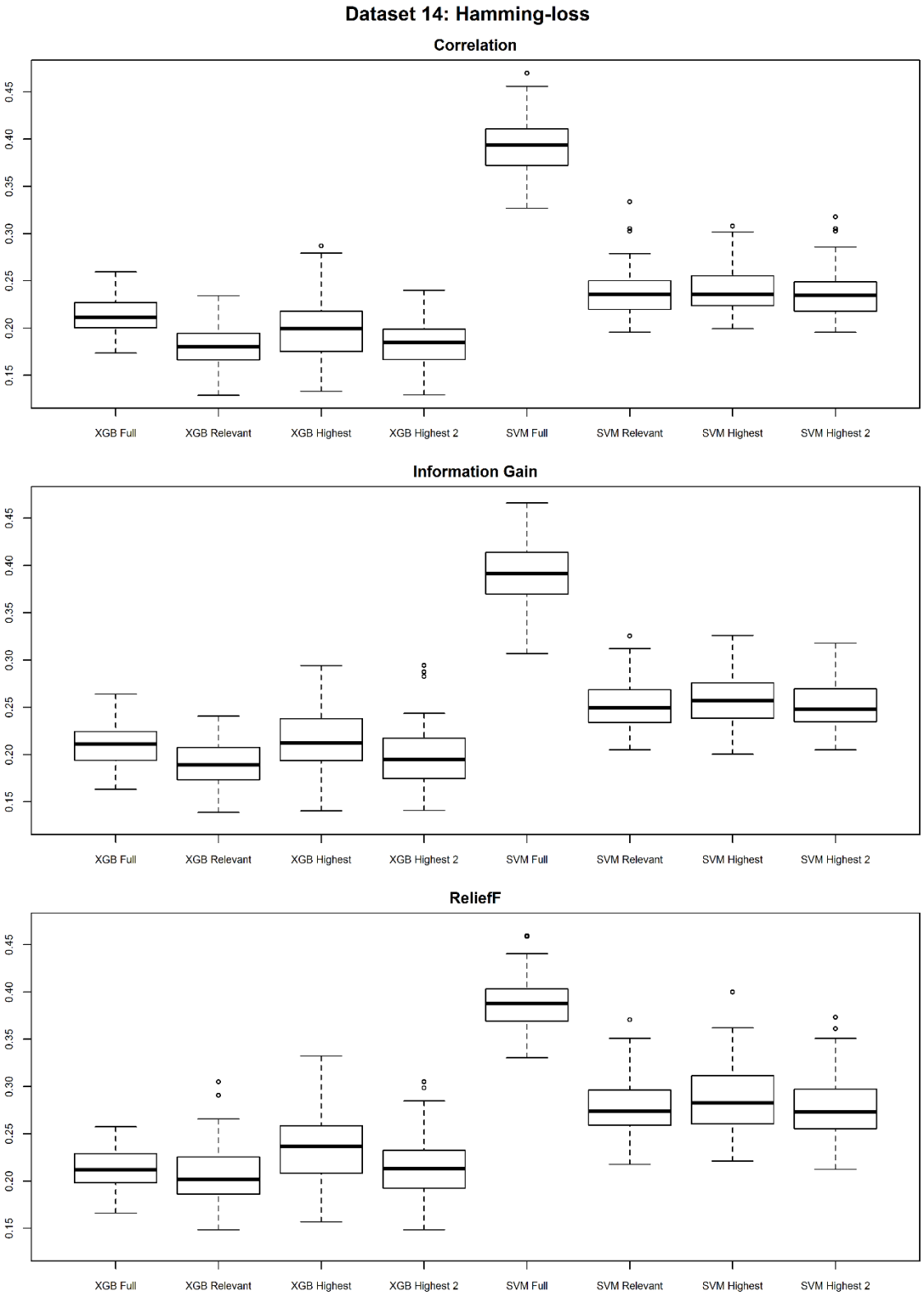


Figure F.53 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 14.

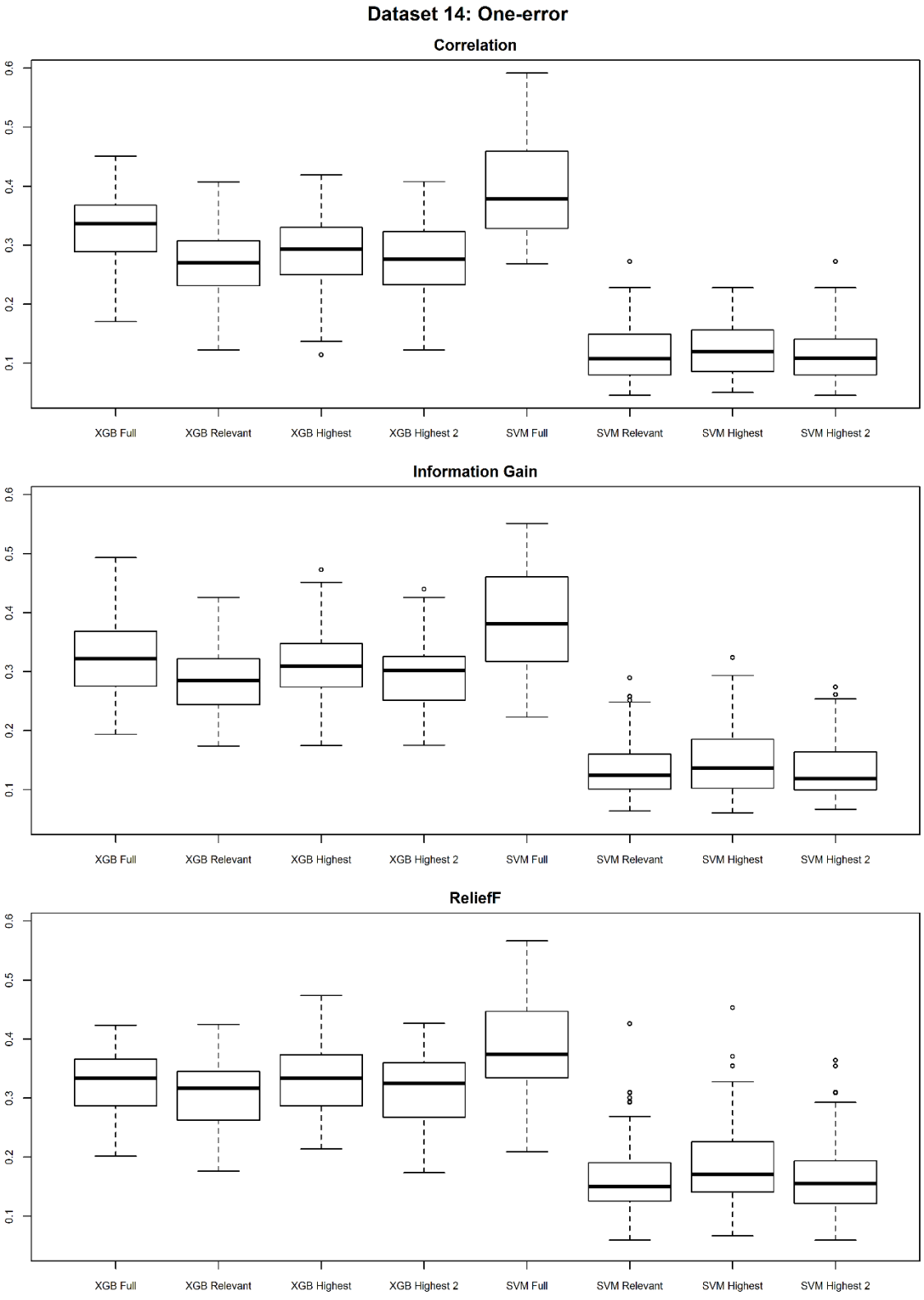


Figure F.54 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 14.

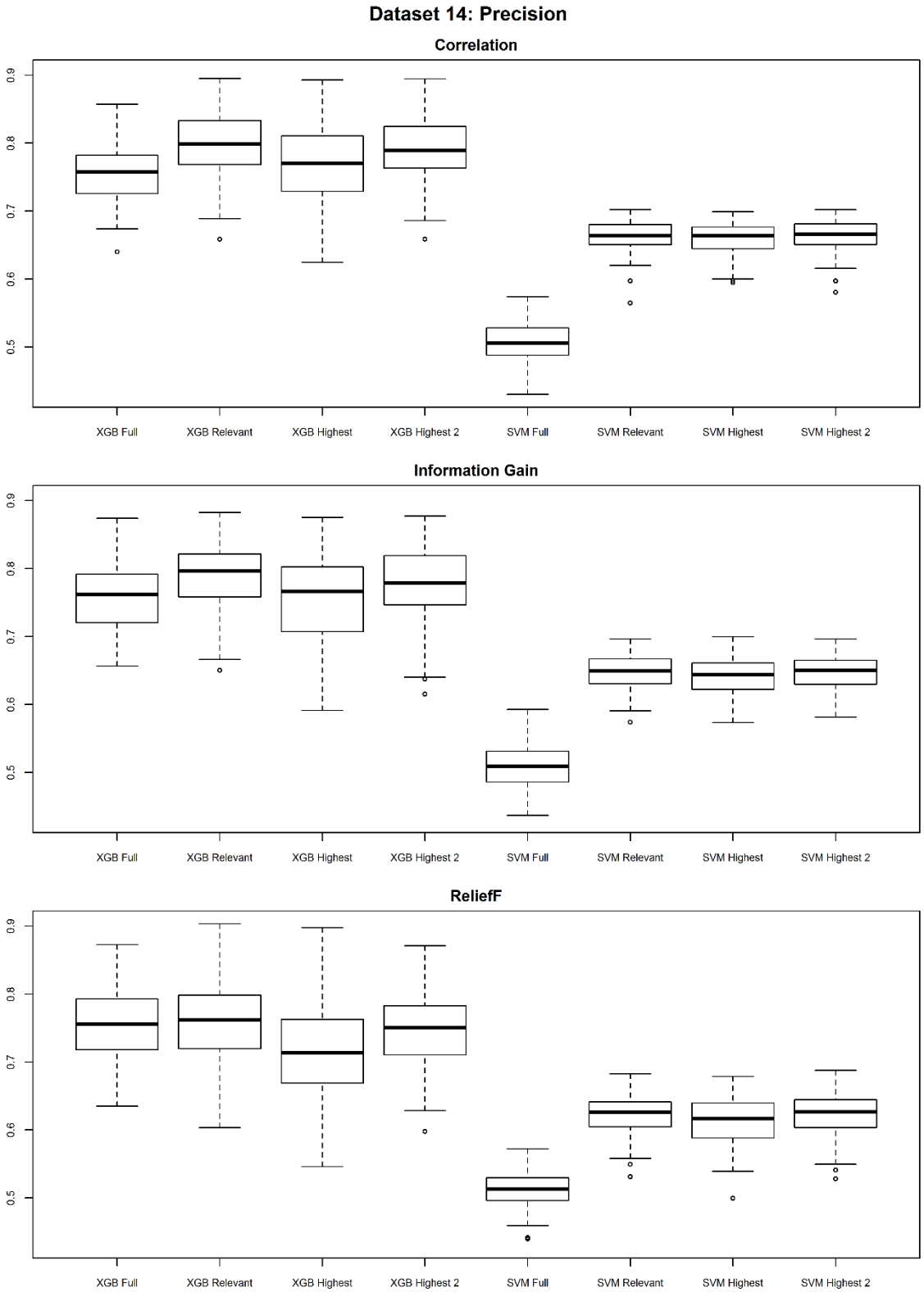


Figure F.55 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 14.

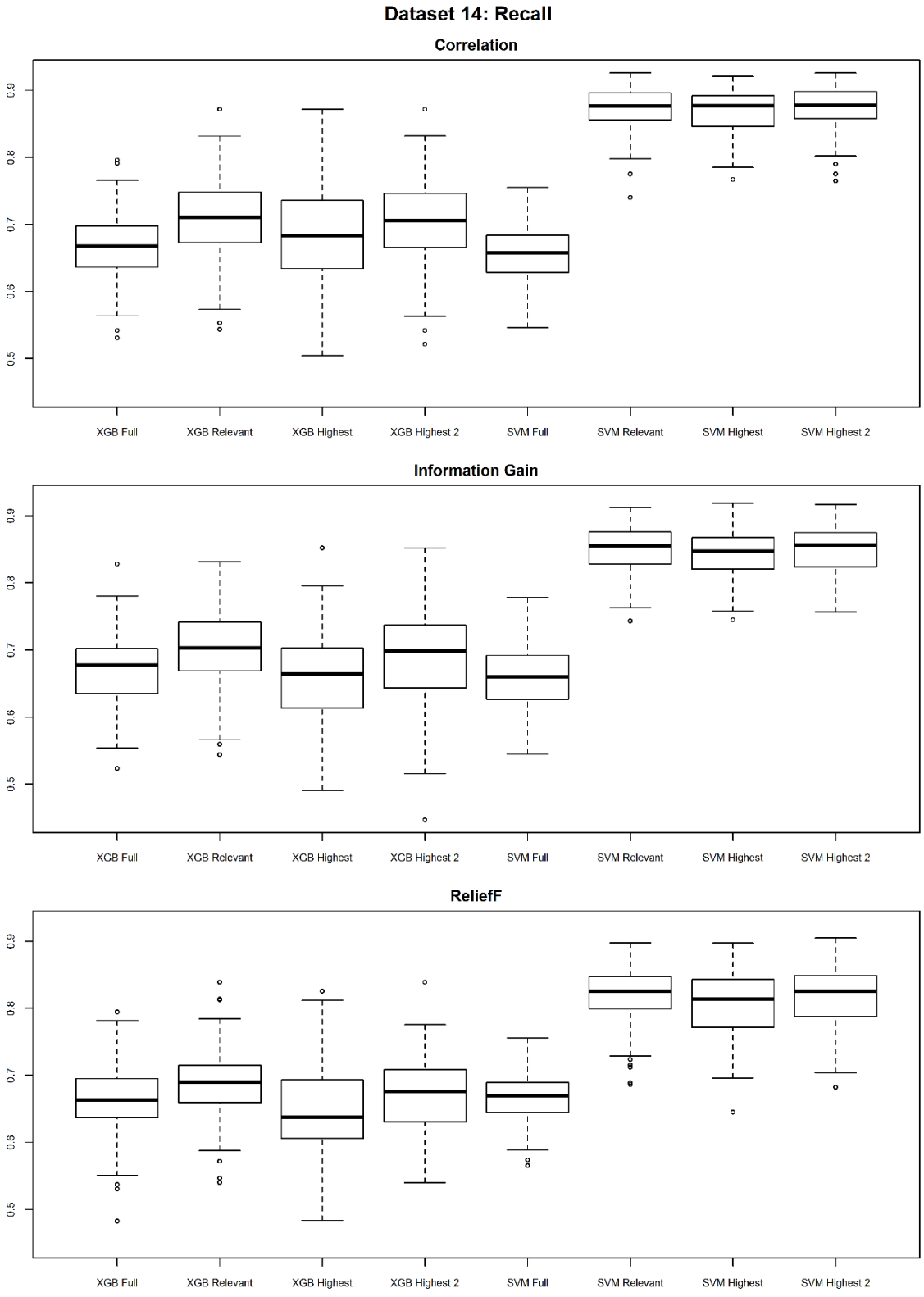


Figure F.56 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 14.

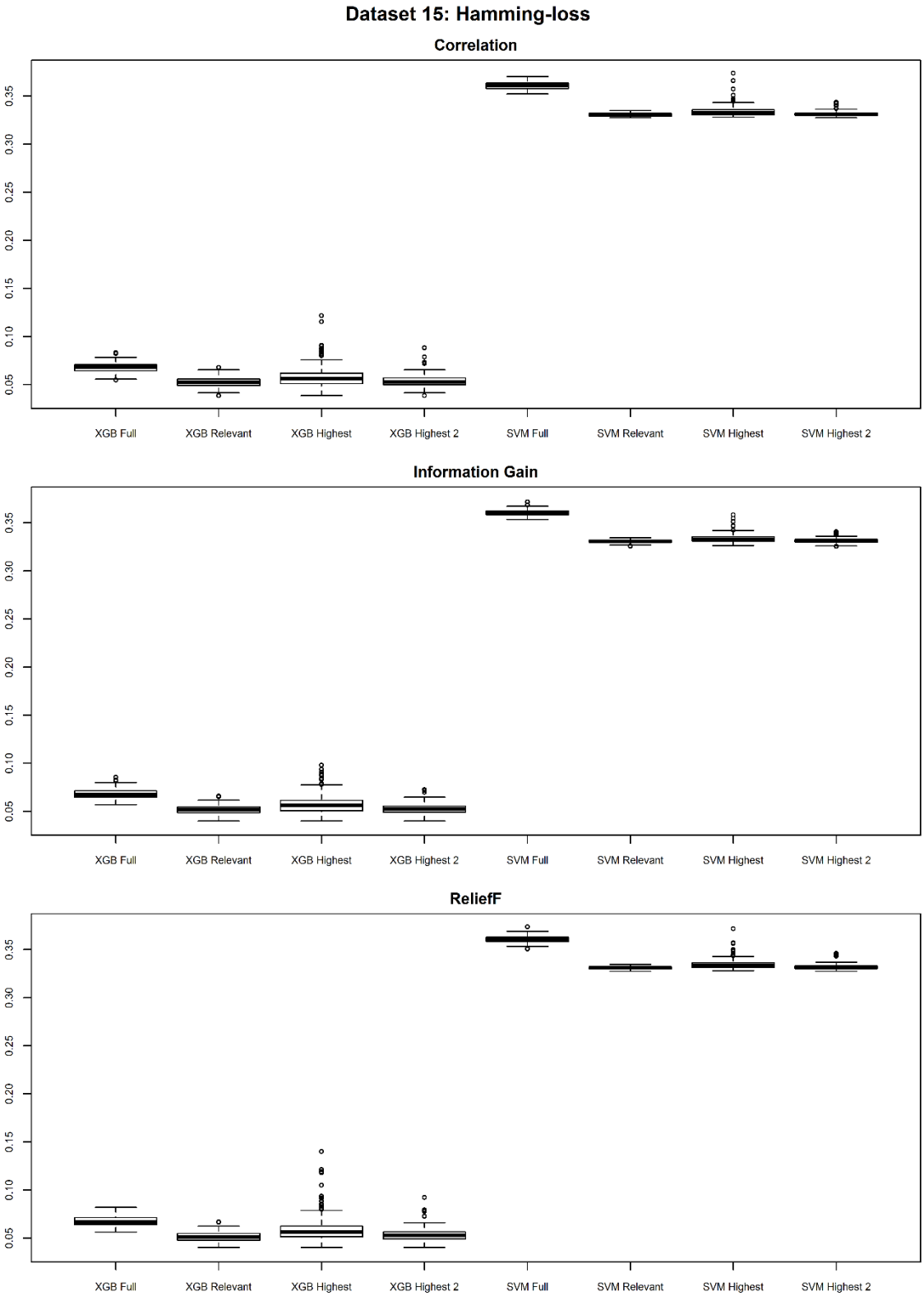


Figure F.57 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 15.

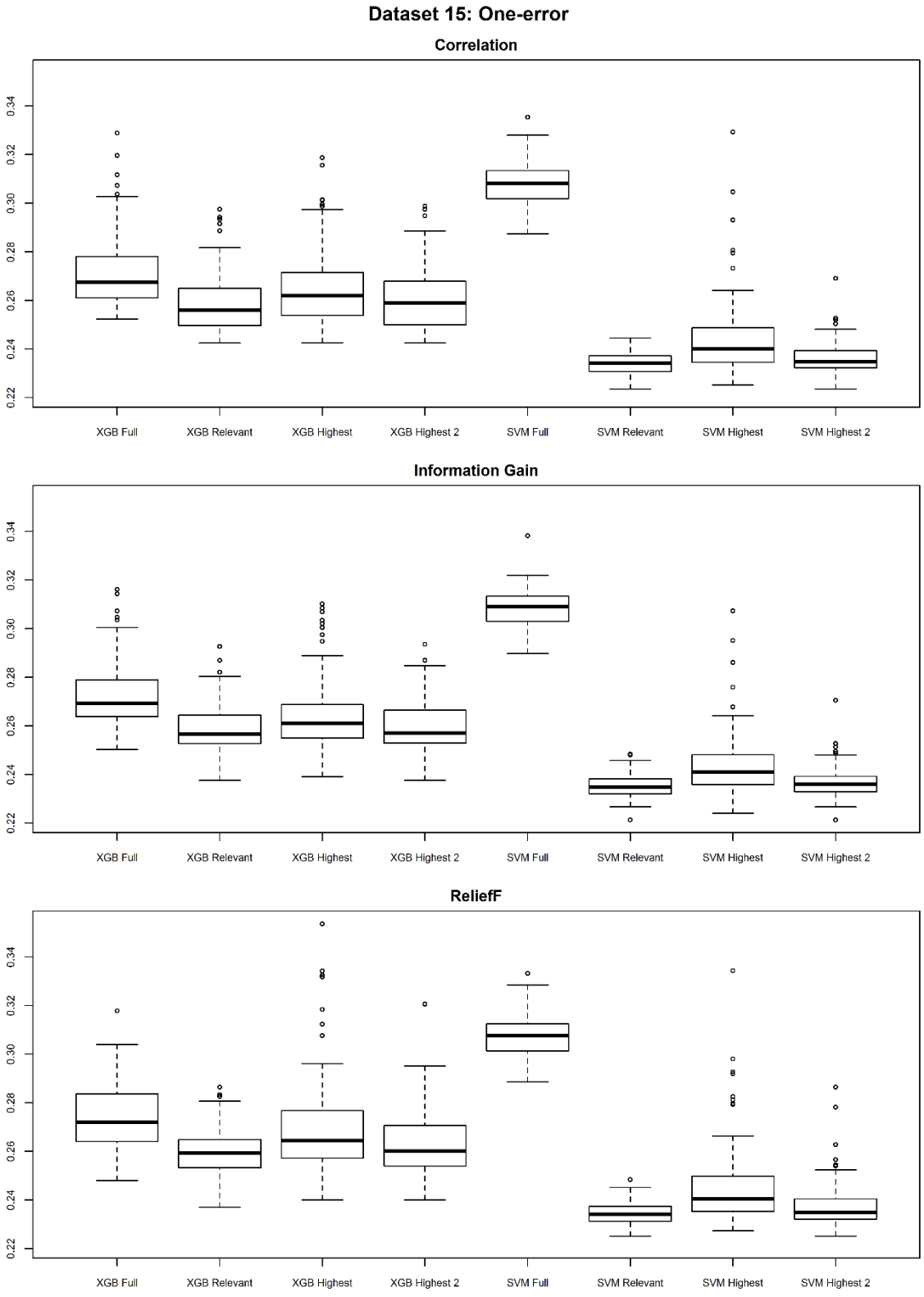


Figure F.58 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 15.

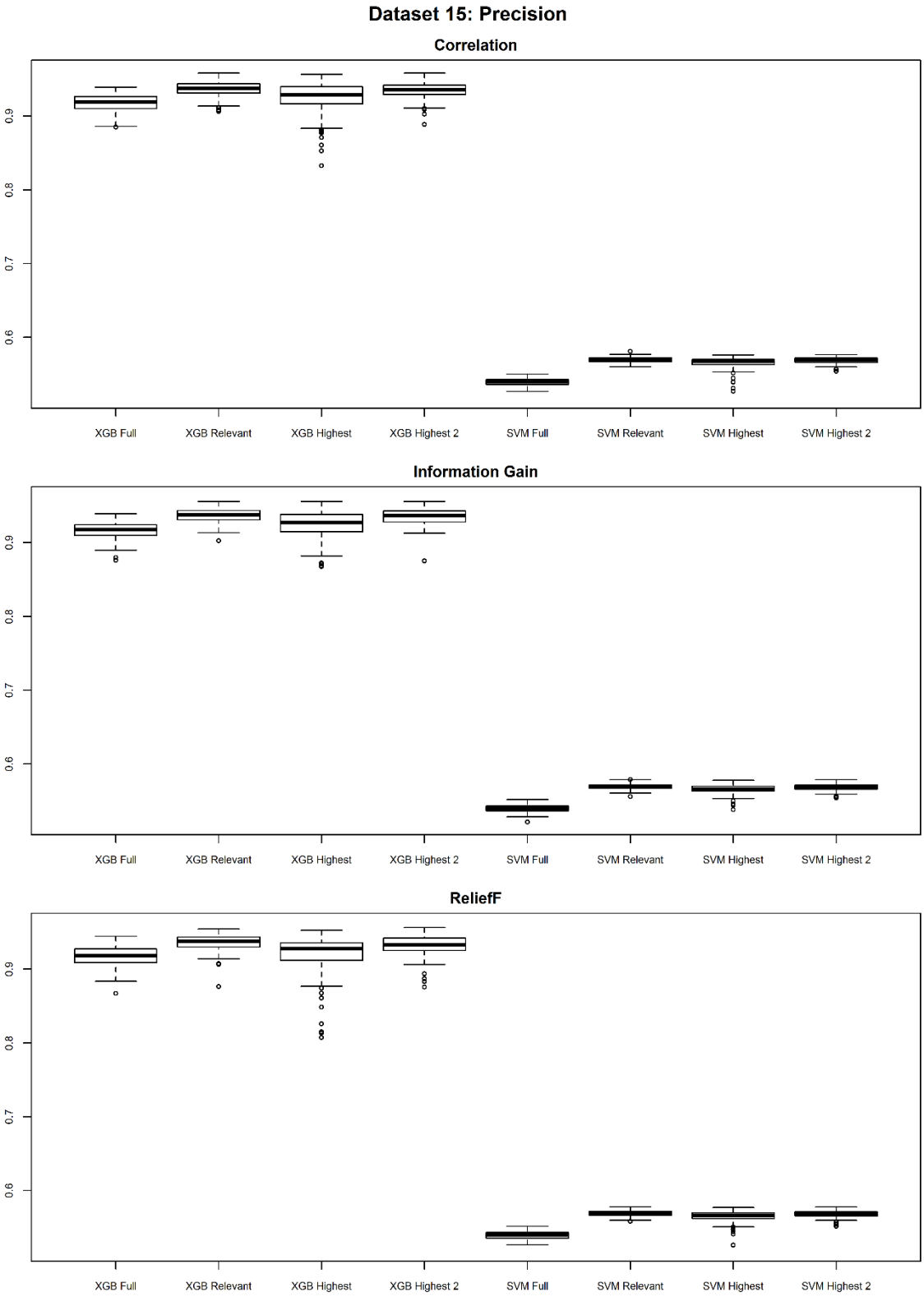


Figure F.59 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 15.

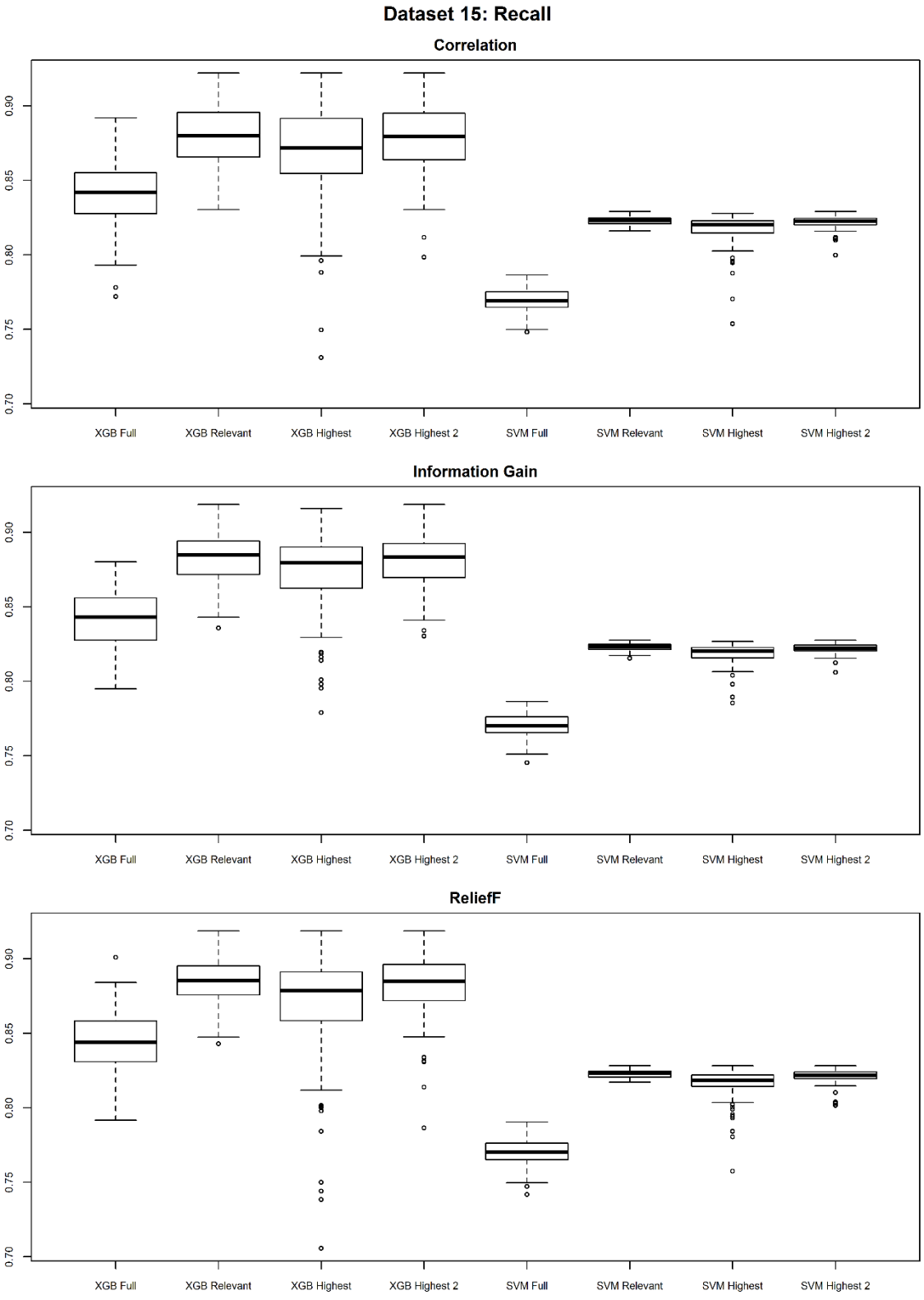


Figure F.60 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 15.

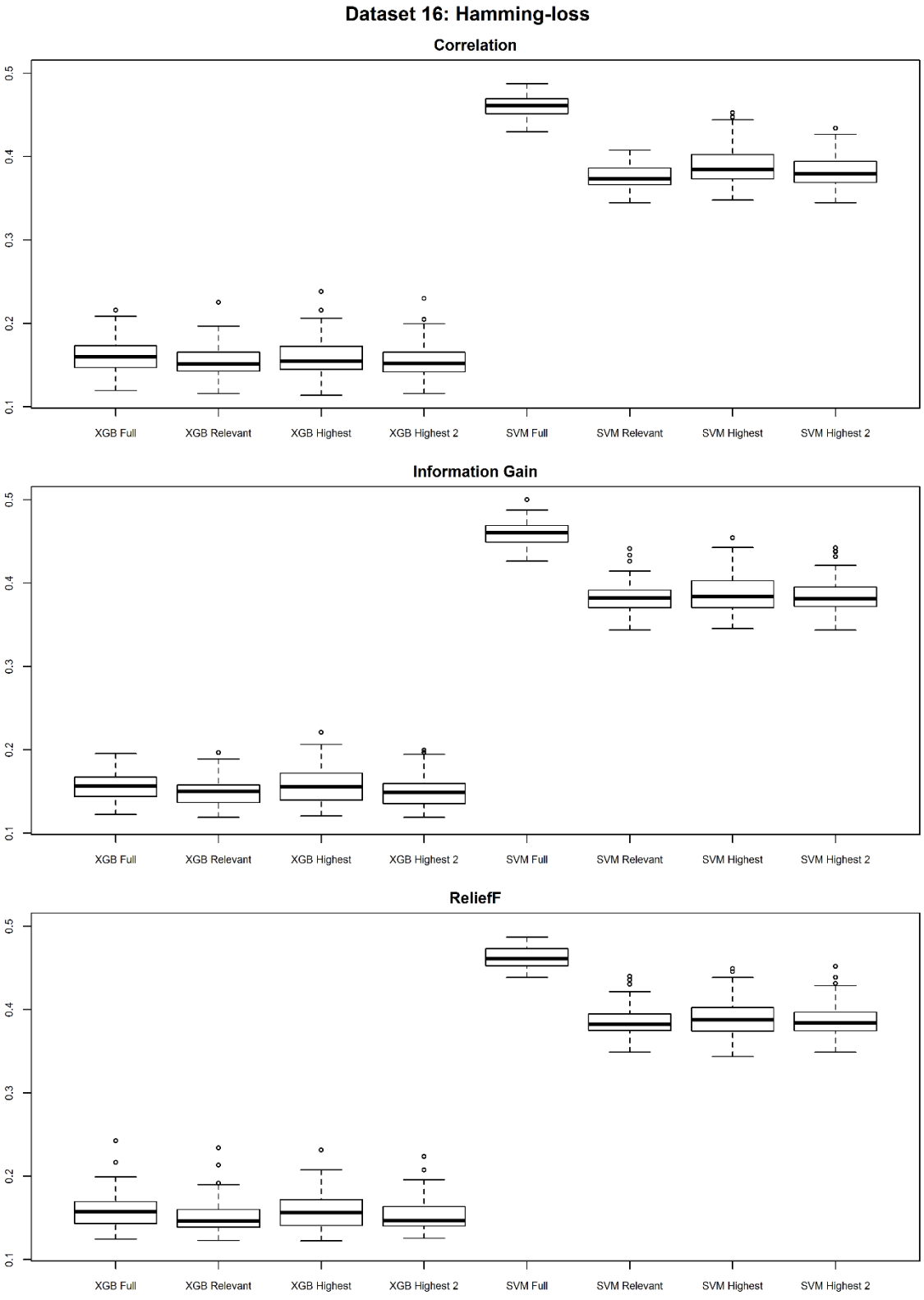


Figure F.61 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 16.

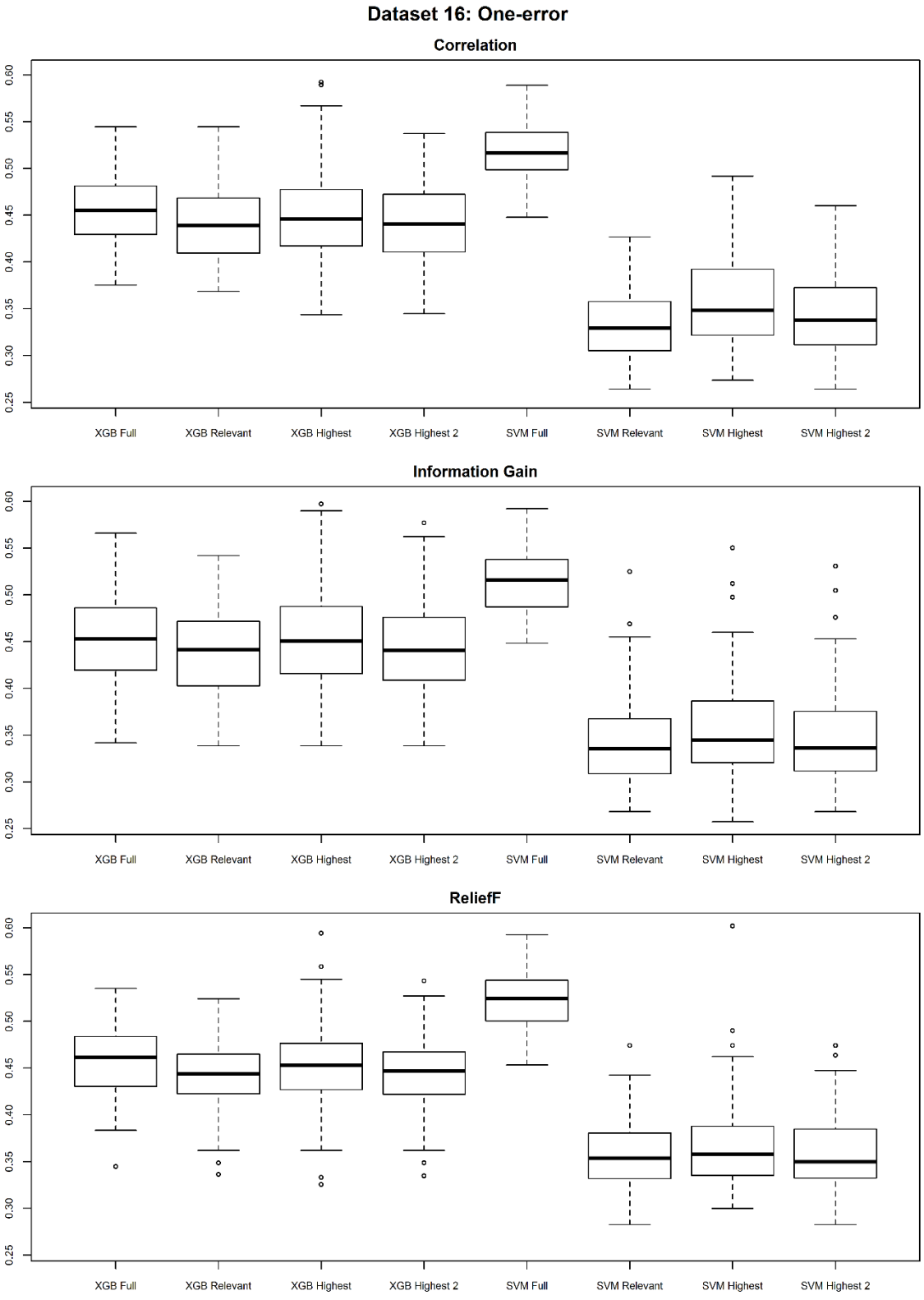


Figure F.62 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 16.

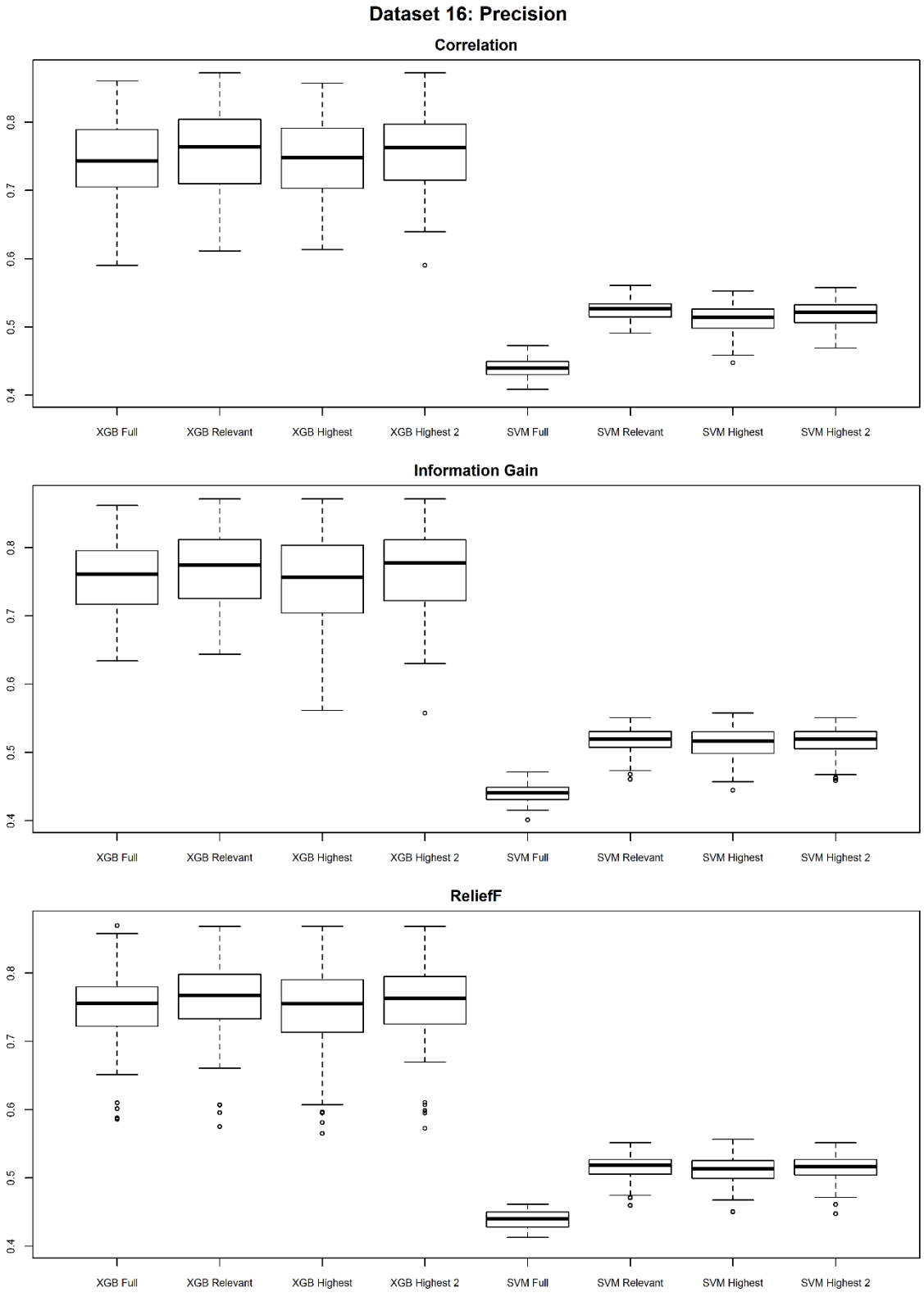


Figure F.63 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 16.

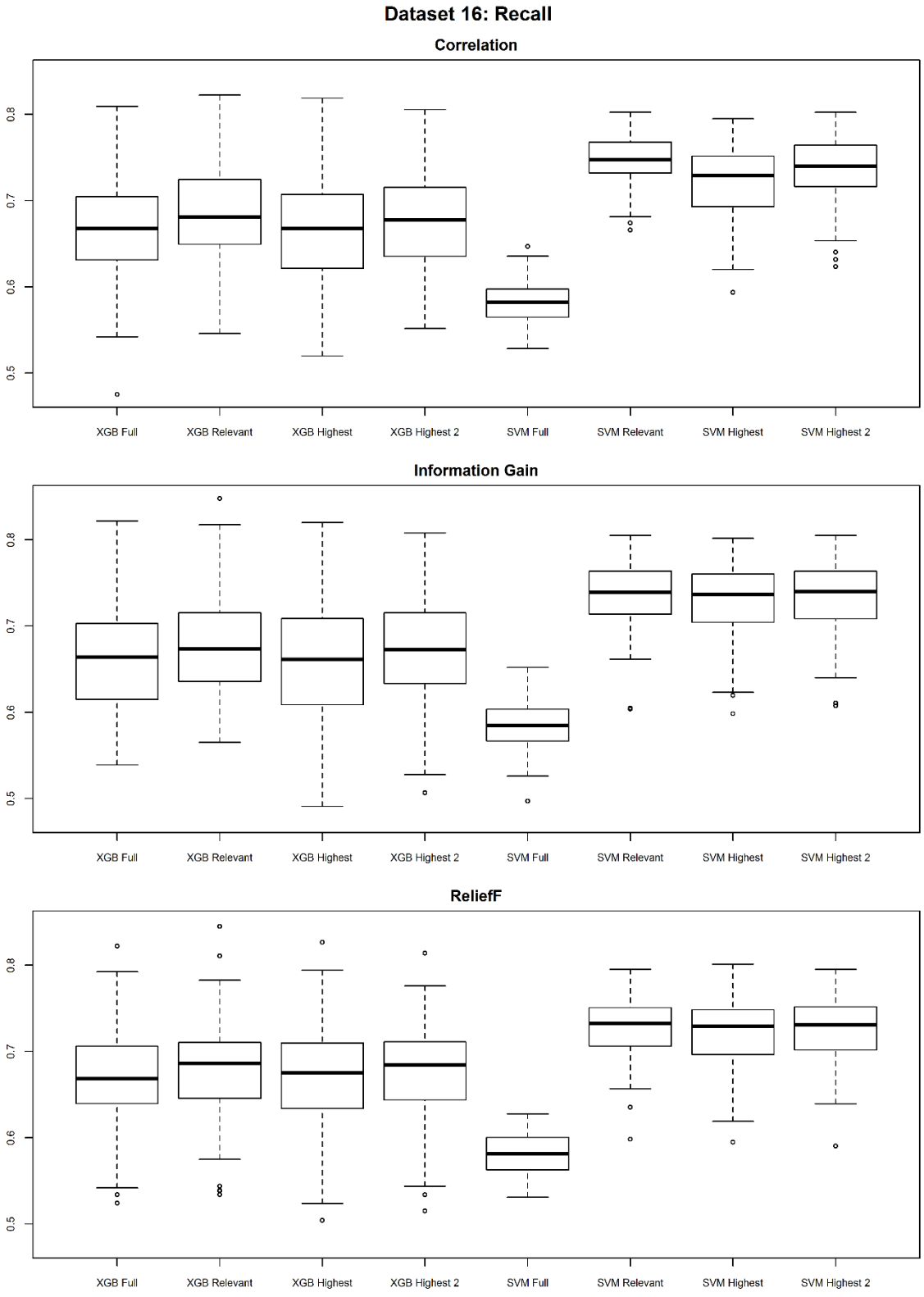


Figure F.64 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 16.

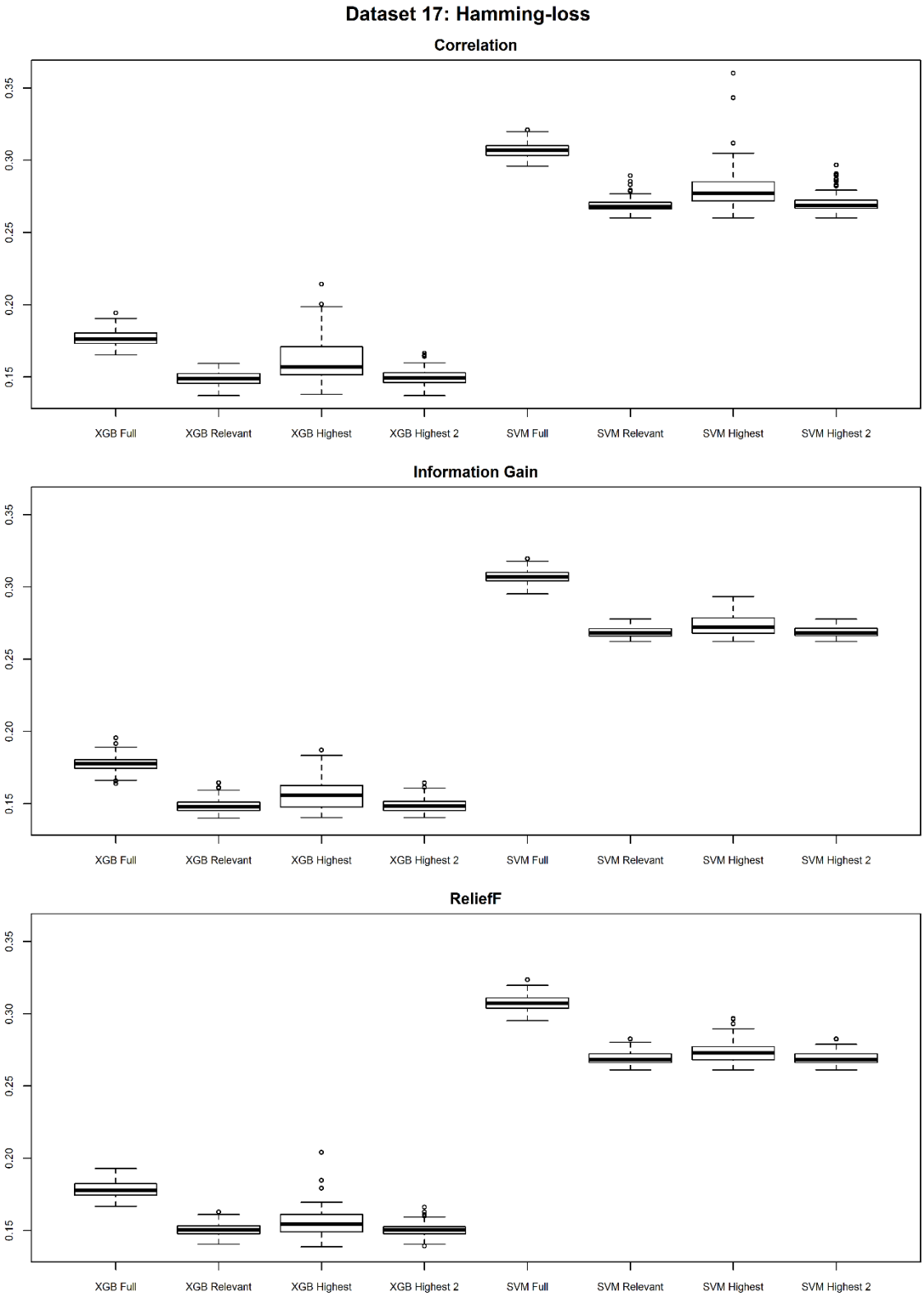


Figure F.65 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 17.

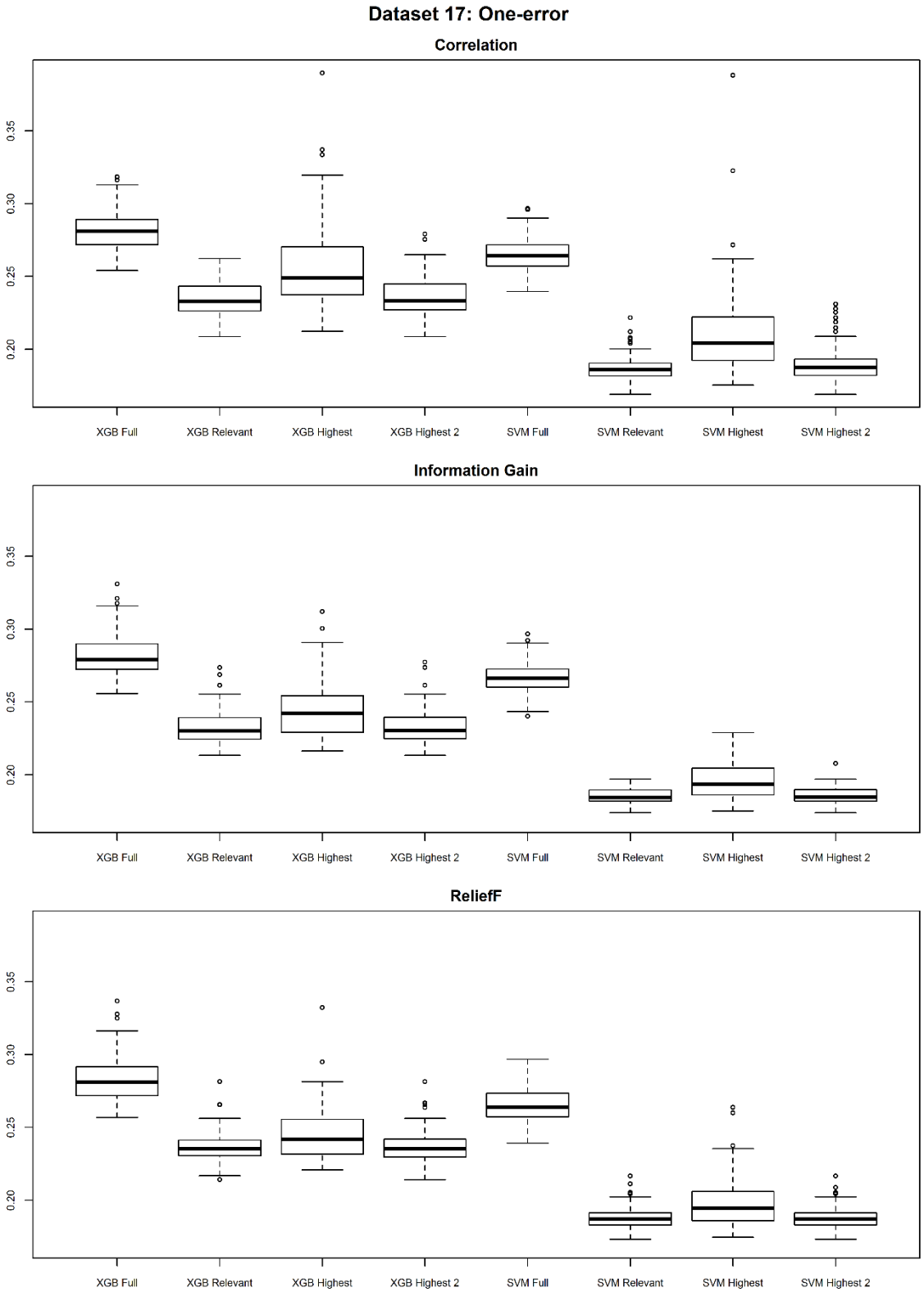


Figure F.66 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 17.

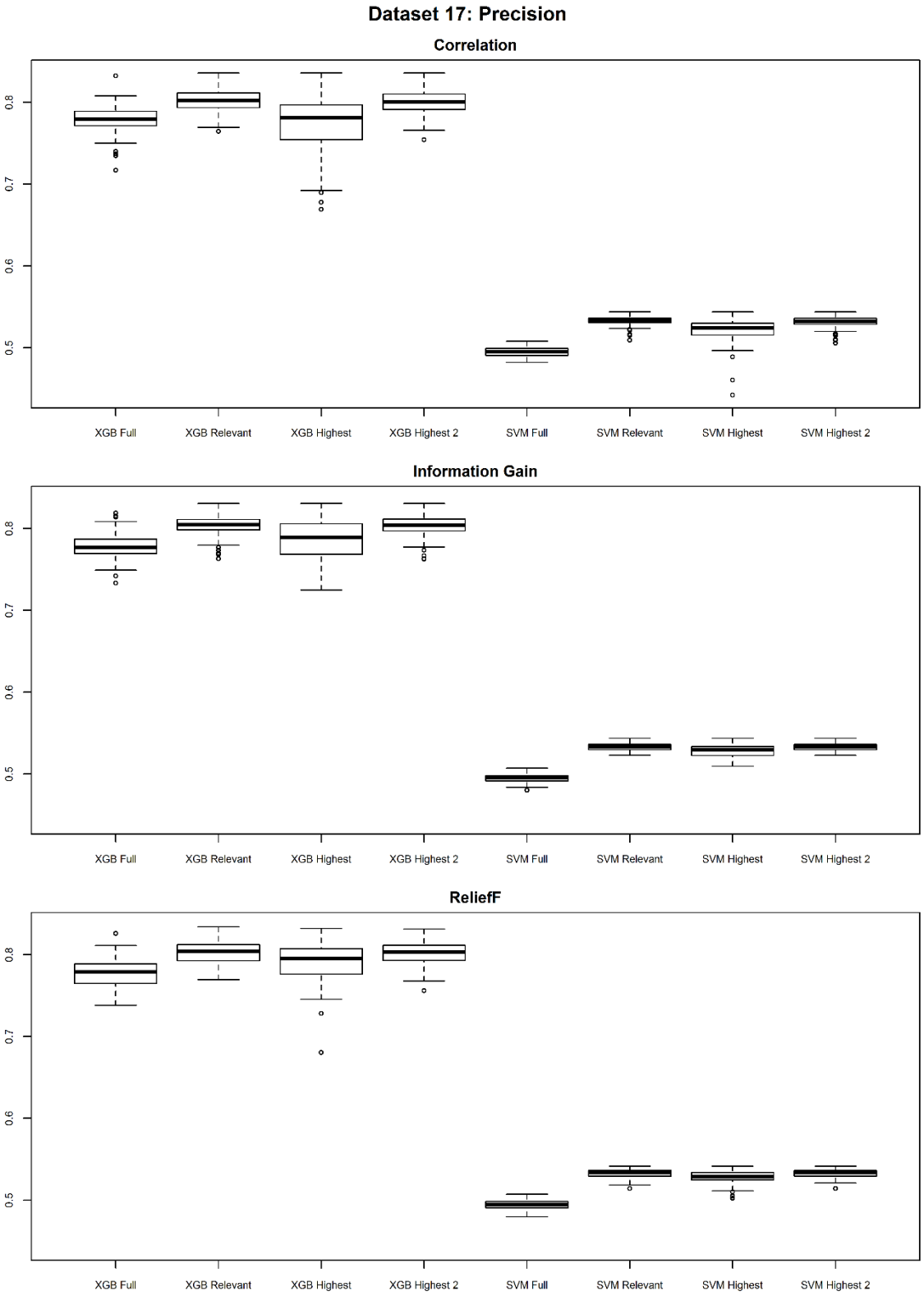


Figure F.67 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 17.

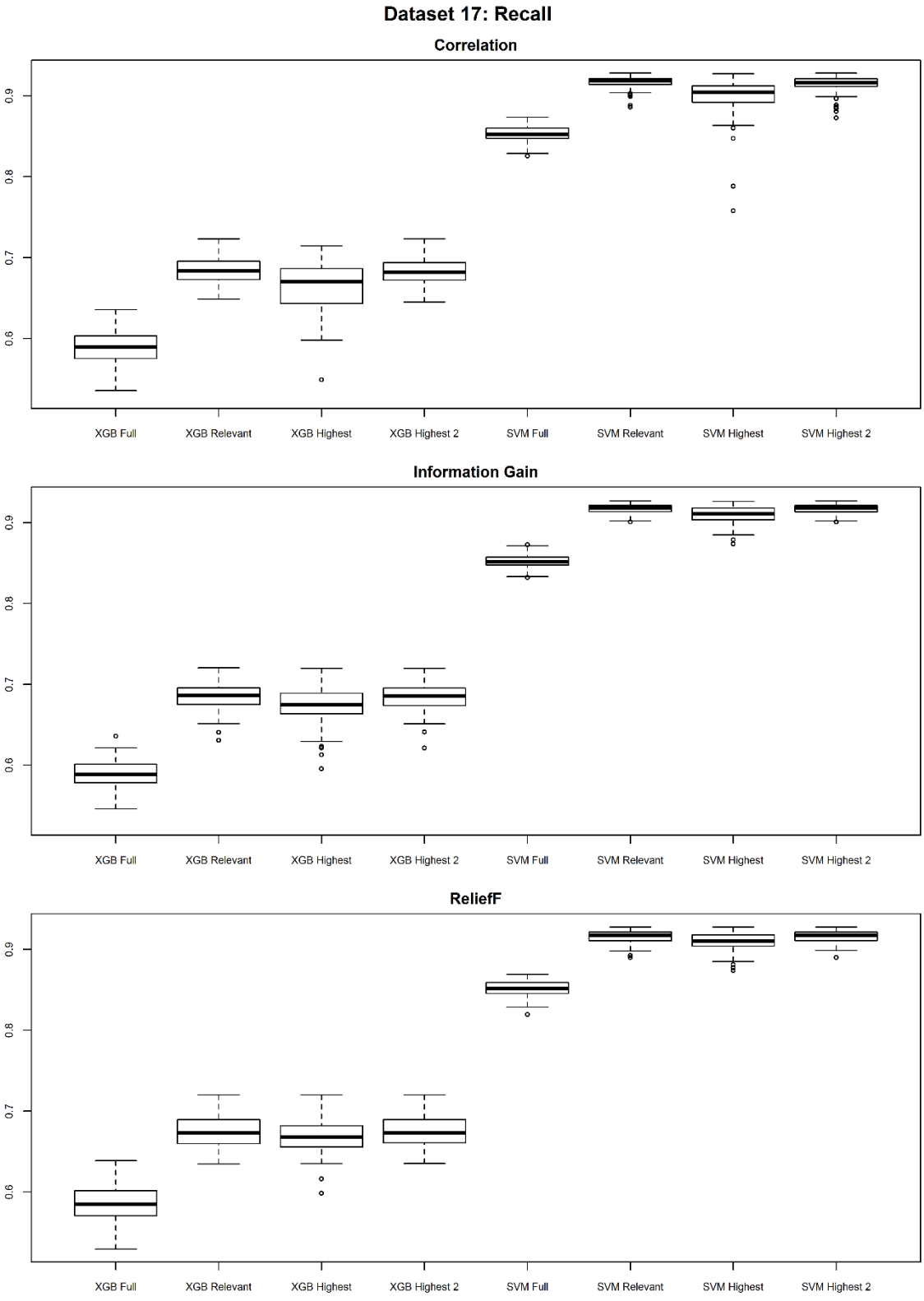


Figure F.68 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 17.

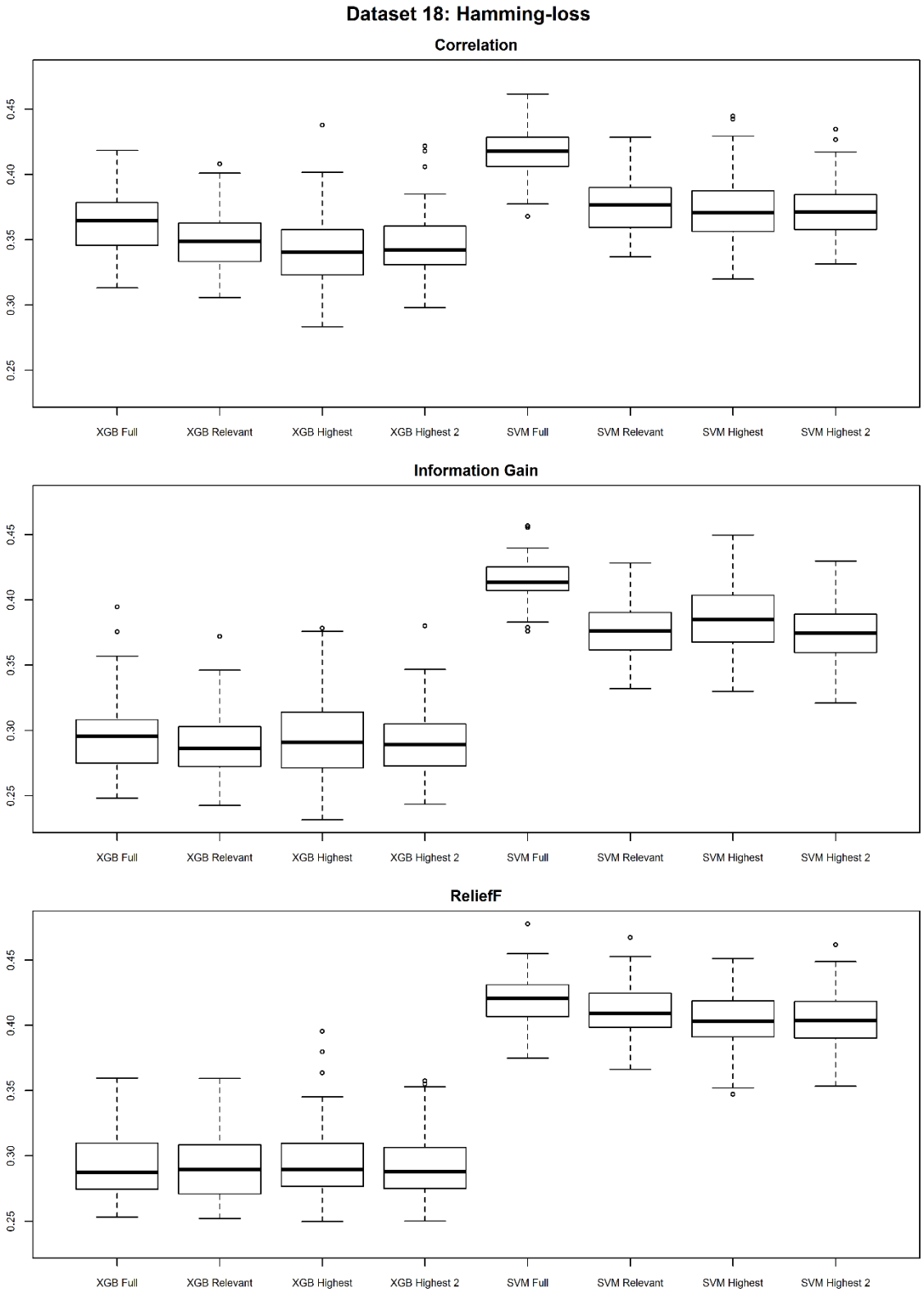


Figure F.69 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 18.

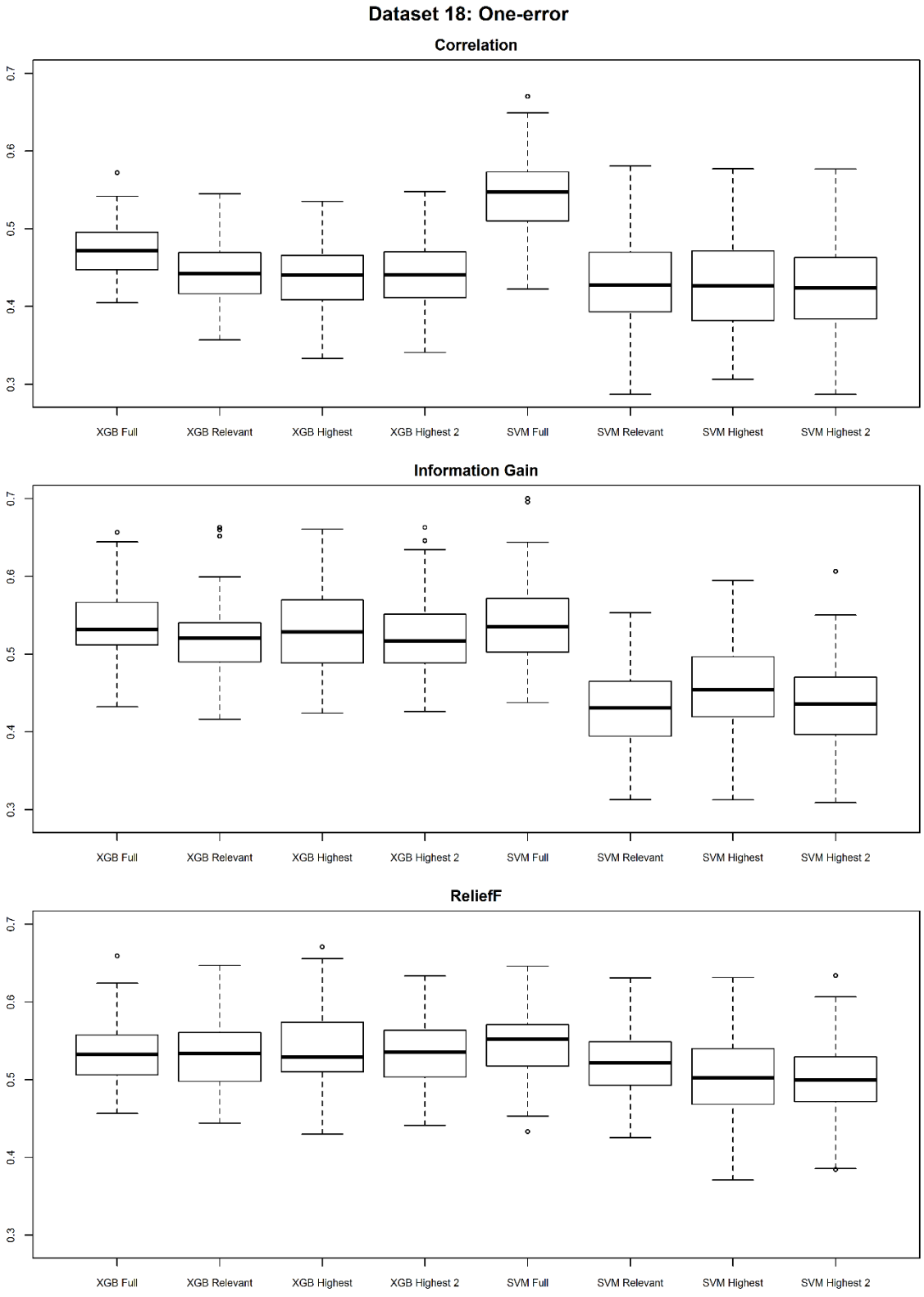


Figure F.70 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 18.

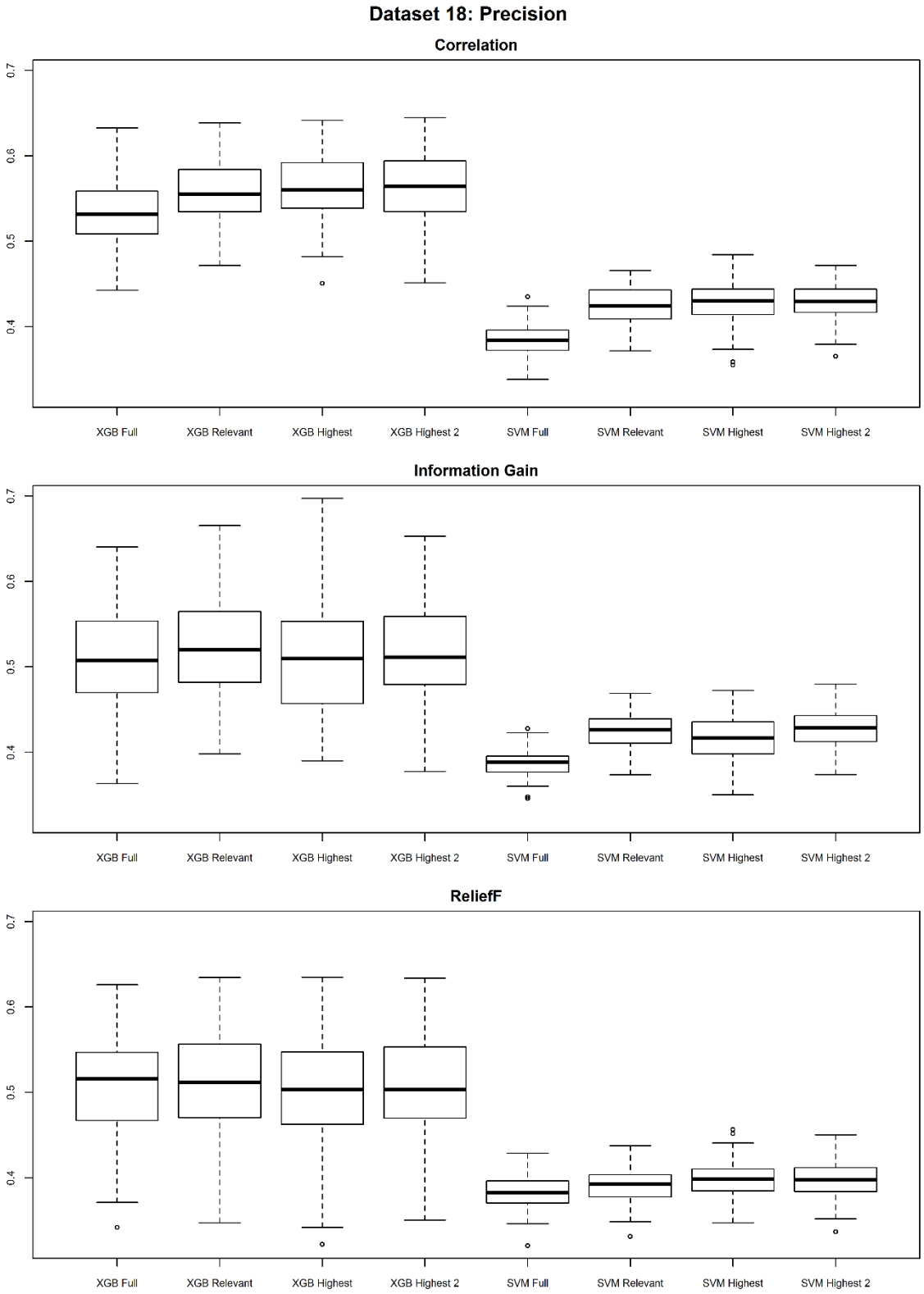


Figure F.71 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 18.

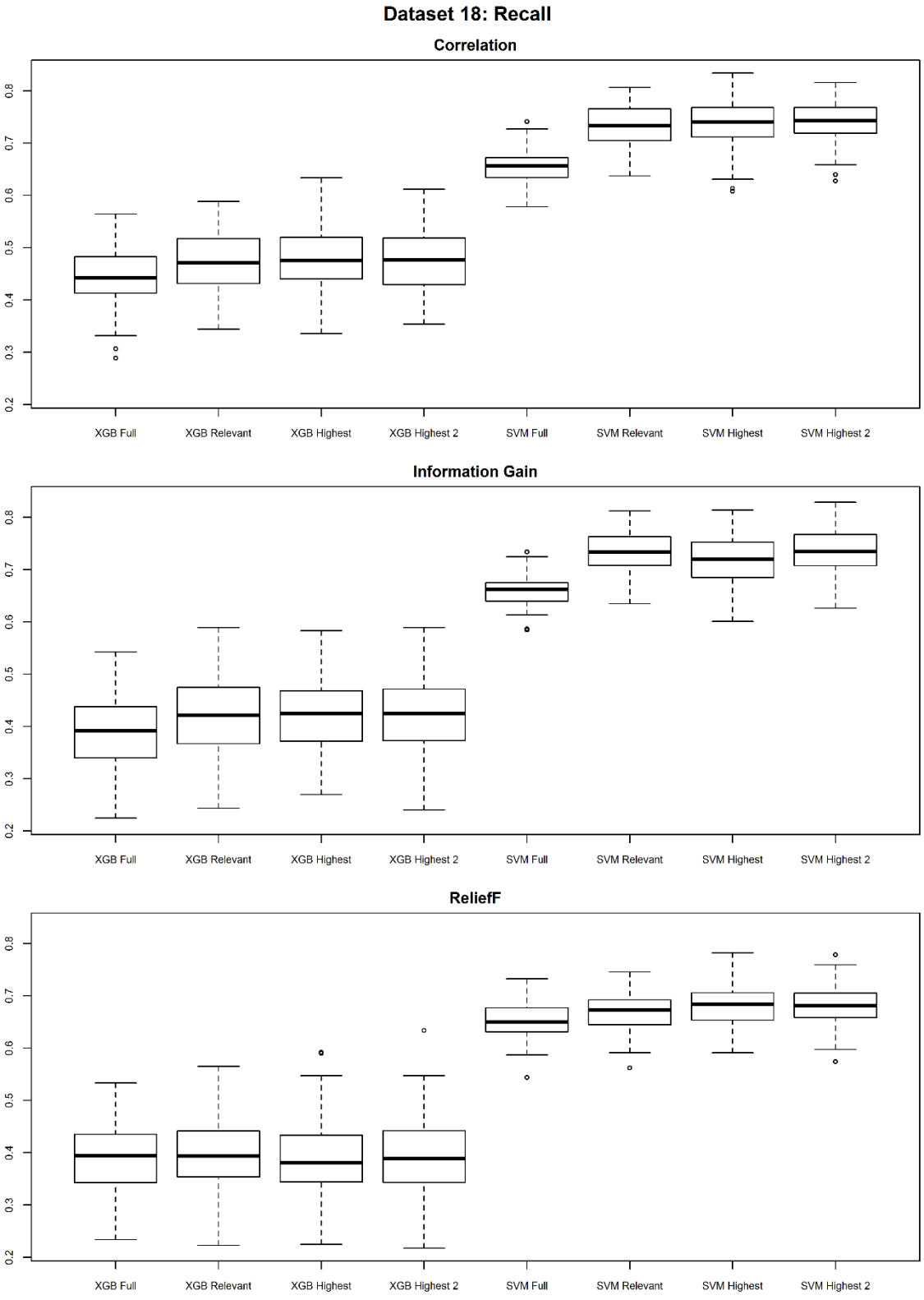


Figure F.72 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 18.

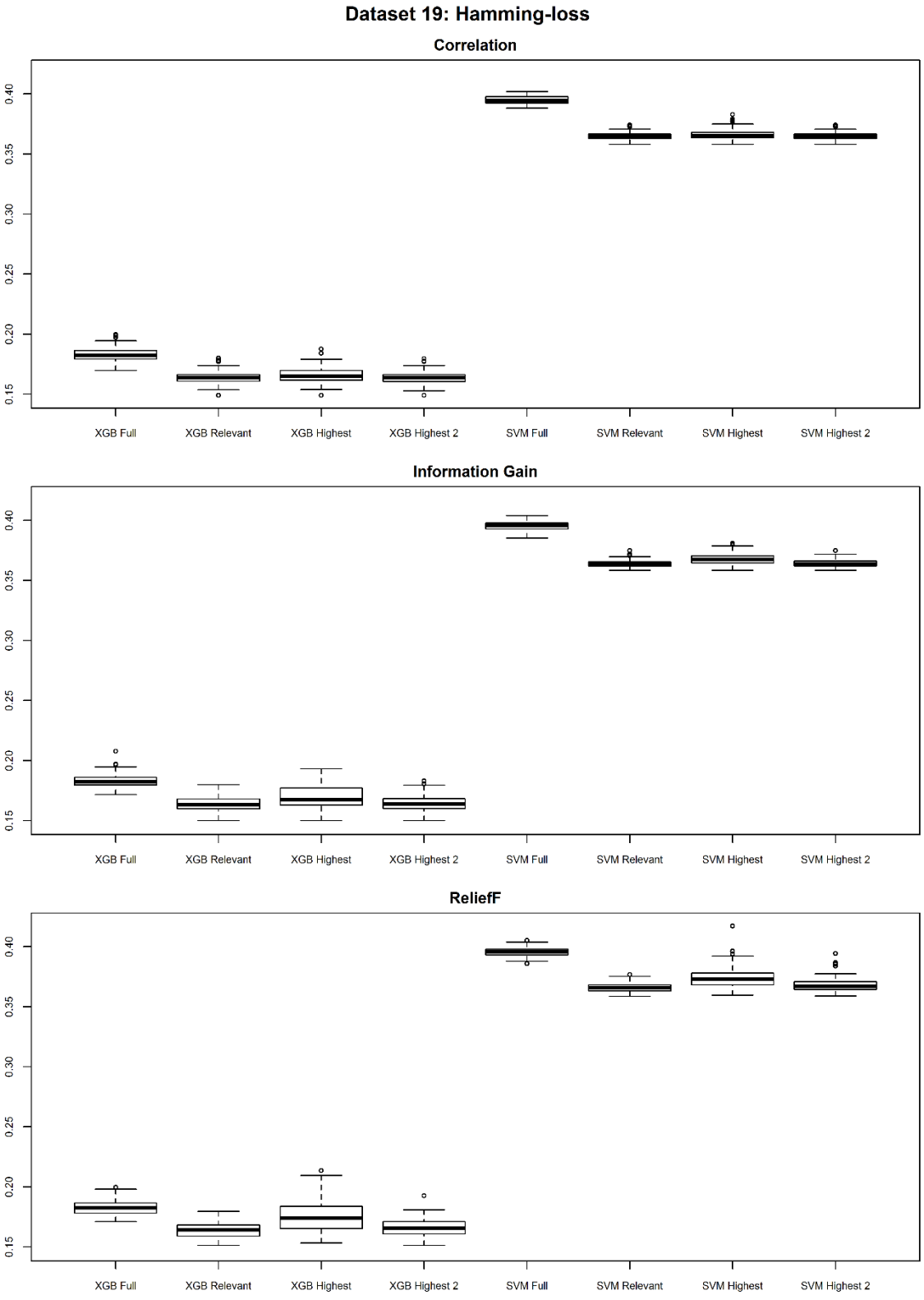


Figure F.73 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 19.

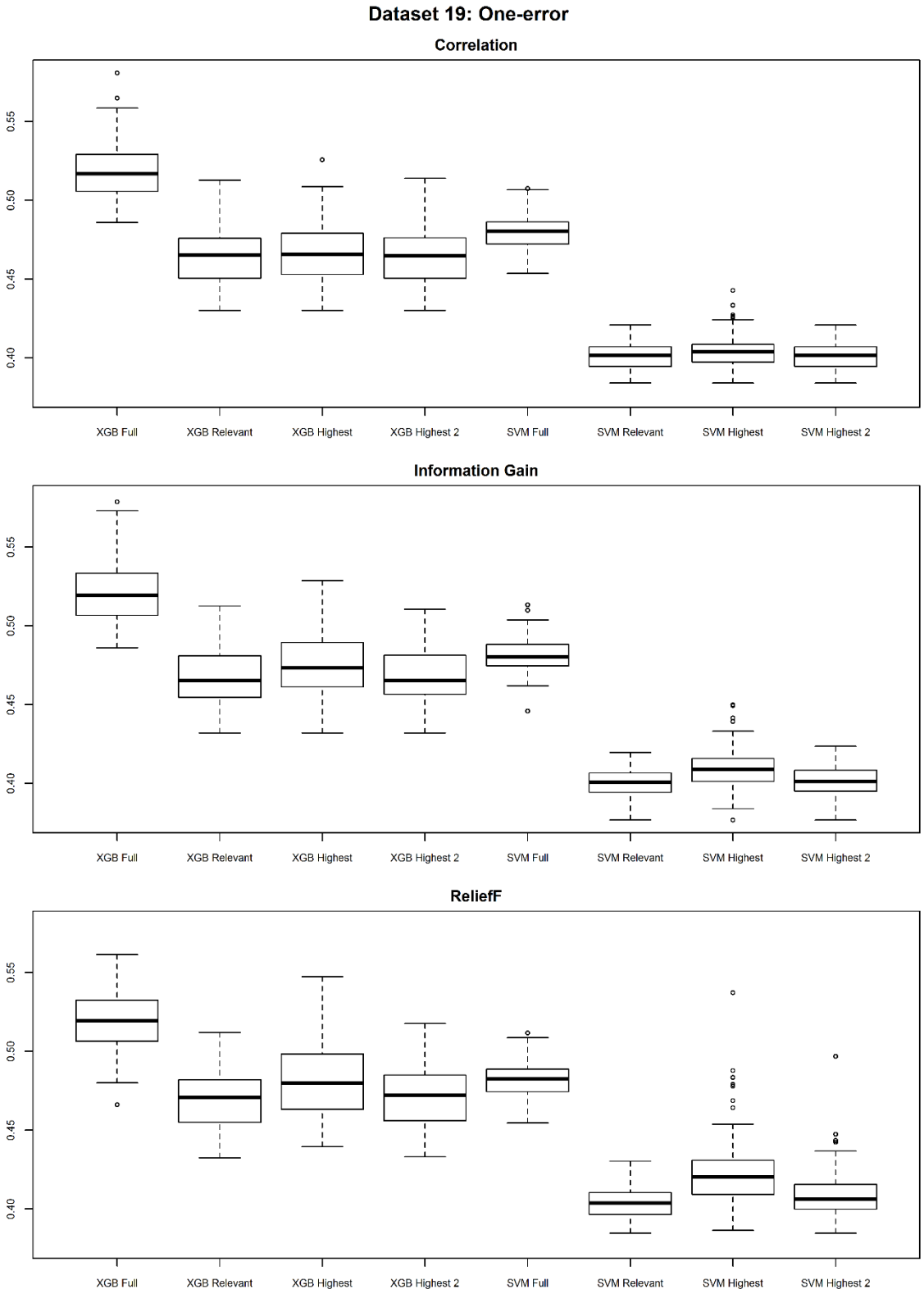


Figure F.74 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 19.

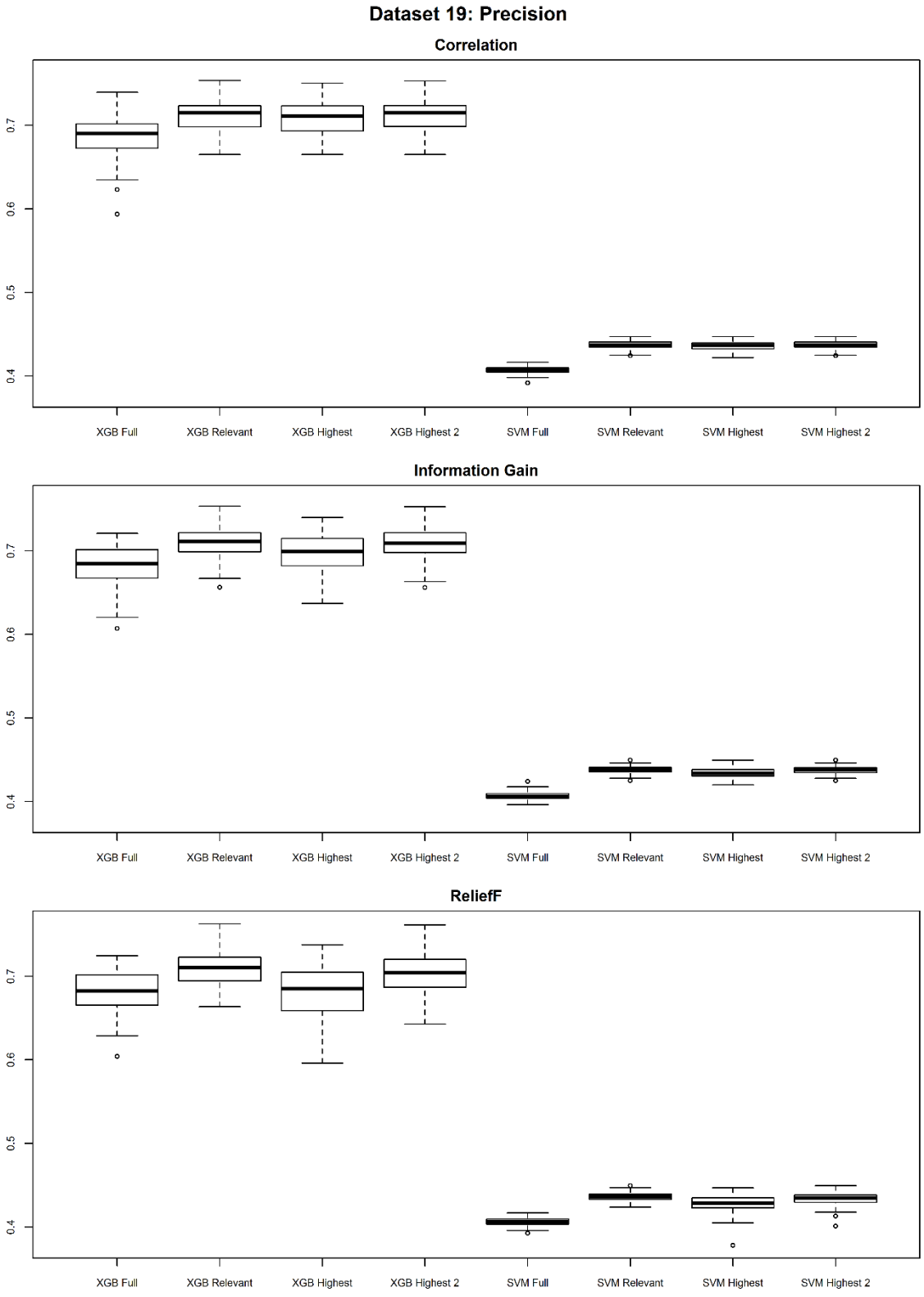


Figure F.75 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 19.

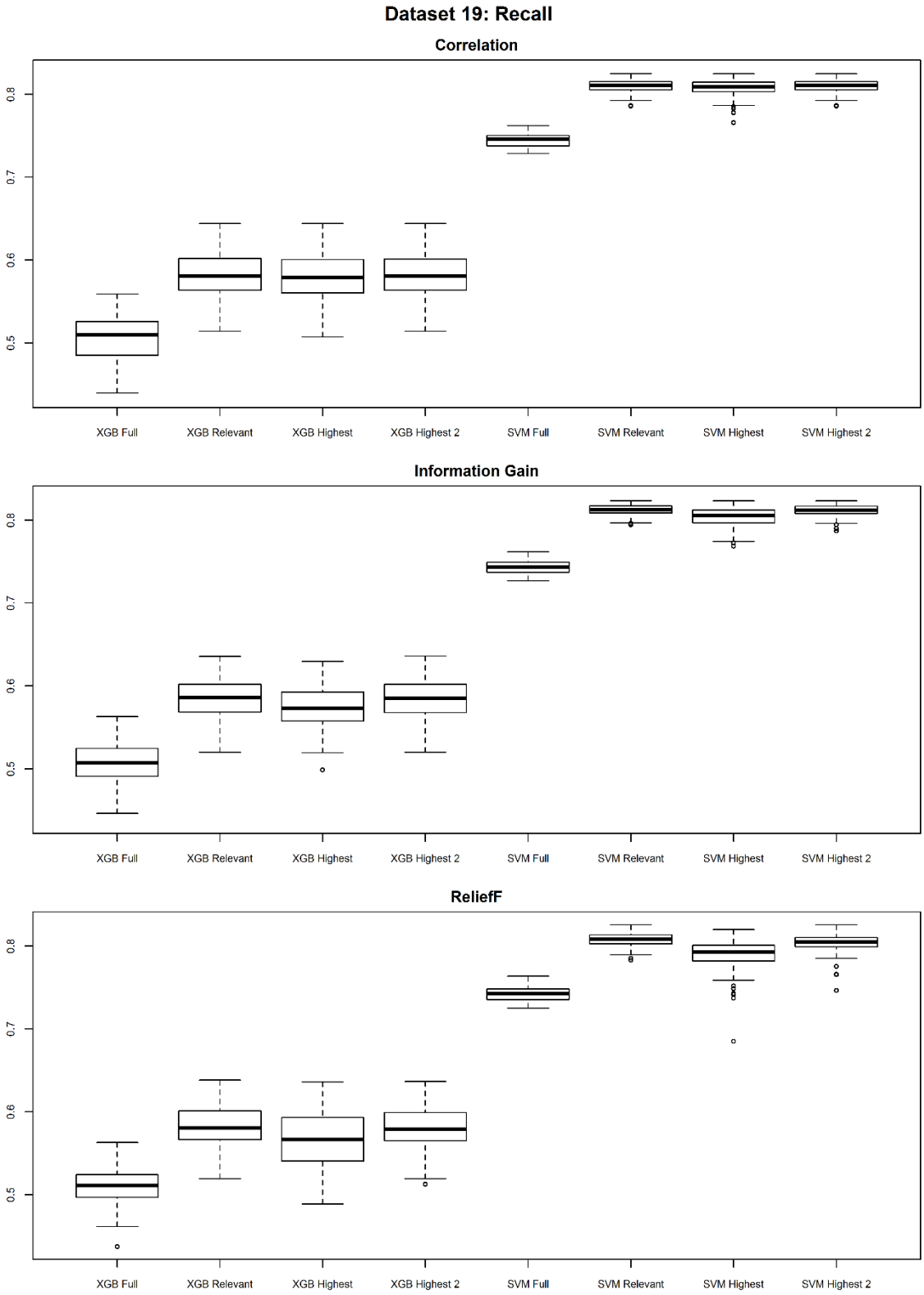


Figure F.76 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 19.

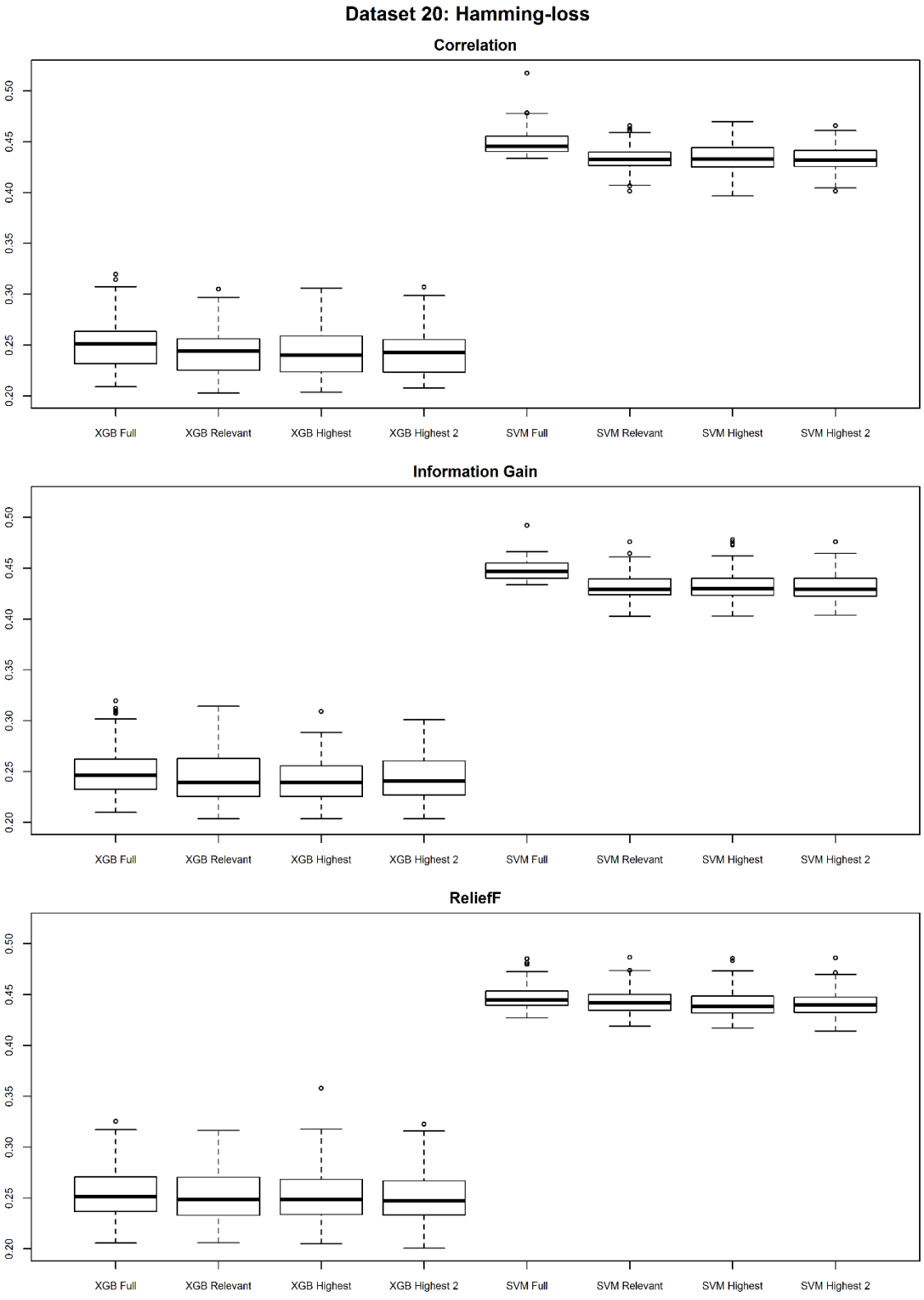


Figure F.77 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 20.

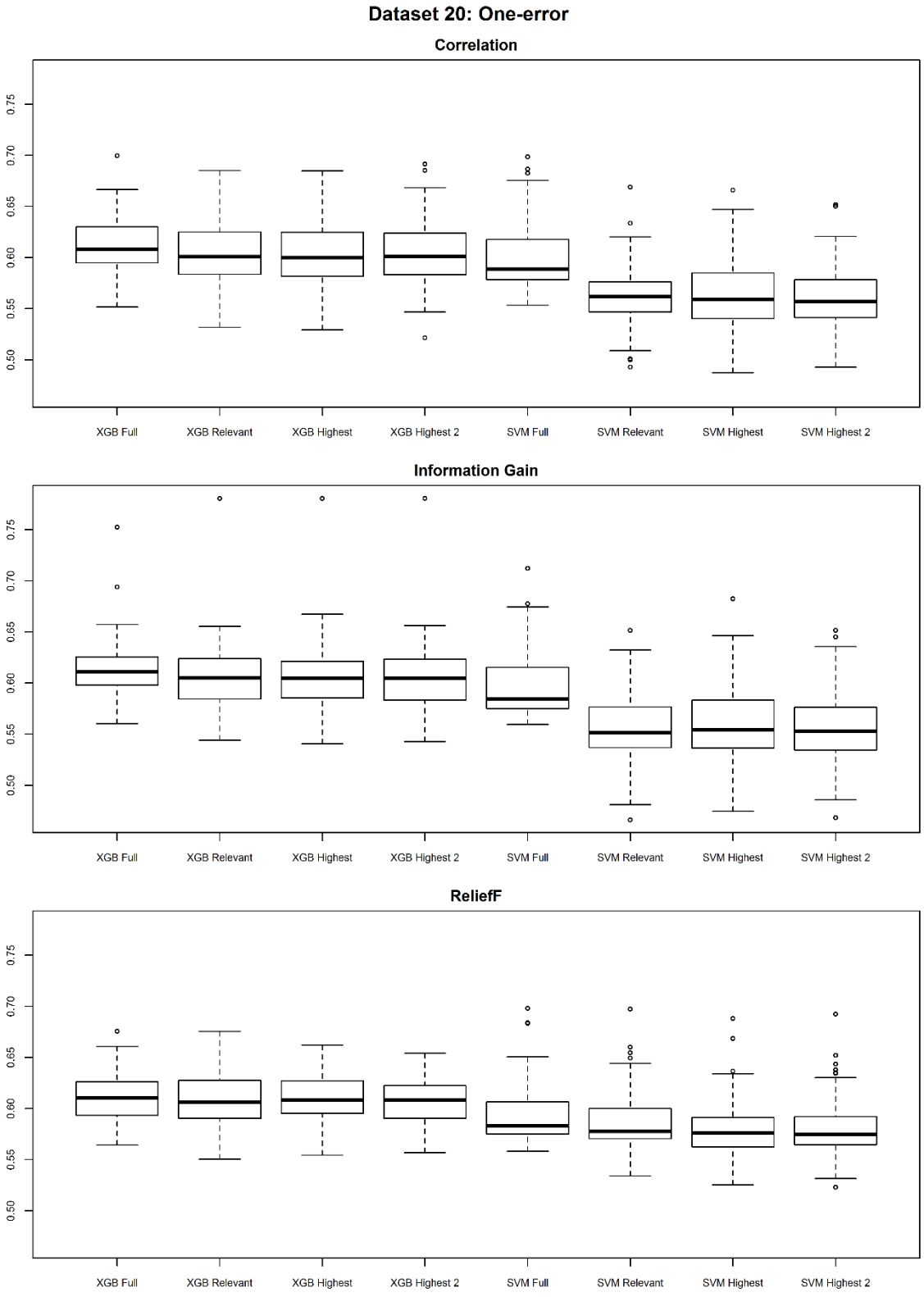


Figure F.78 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 20.

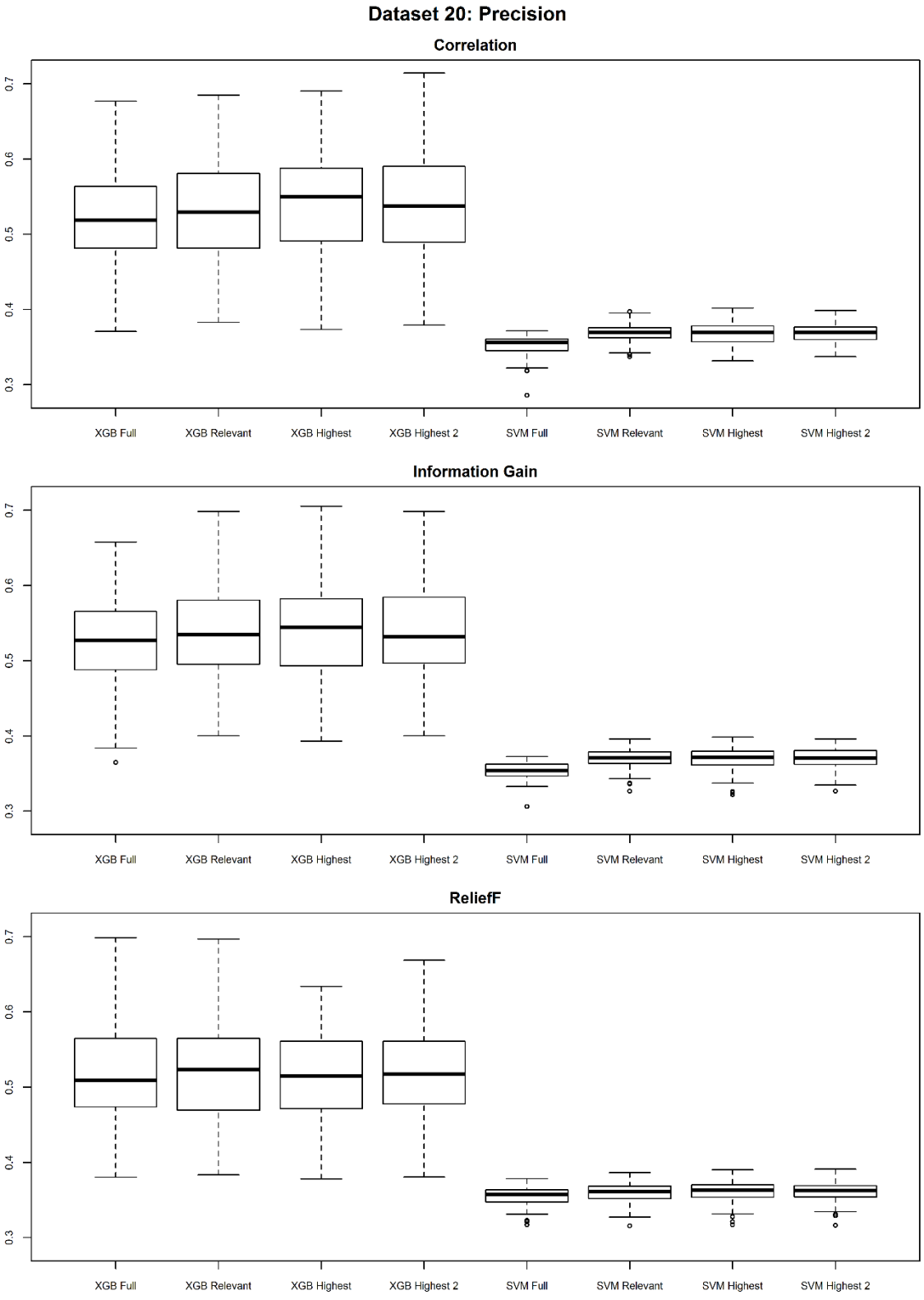


Figure F.79 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 20.

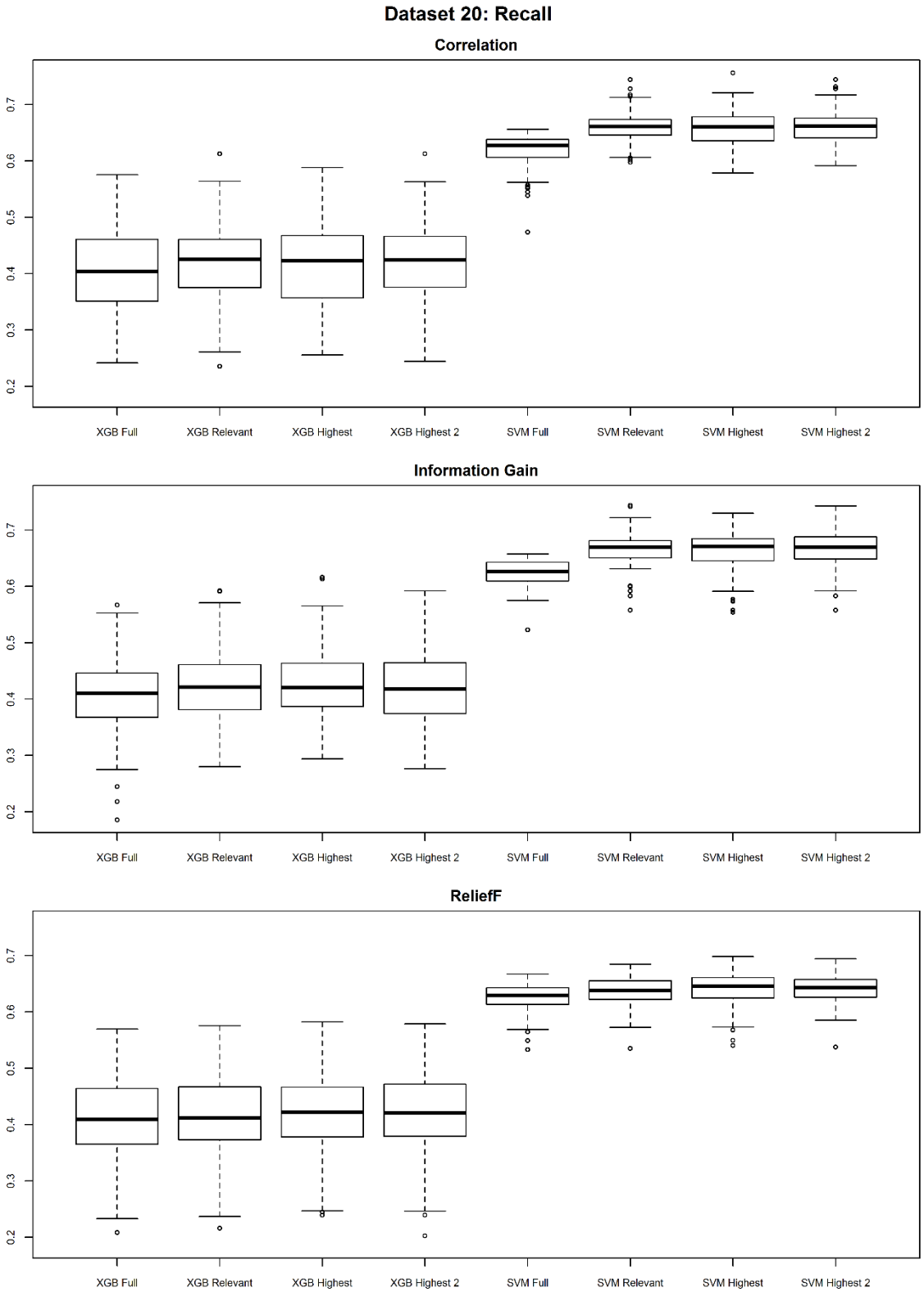


Figure F.80 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 20.

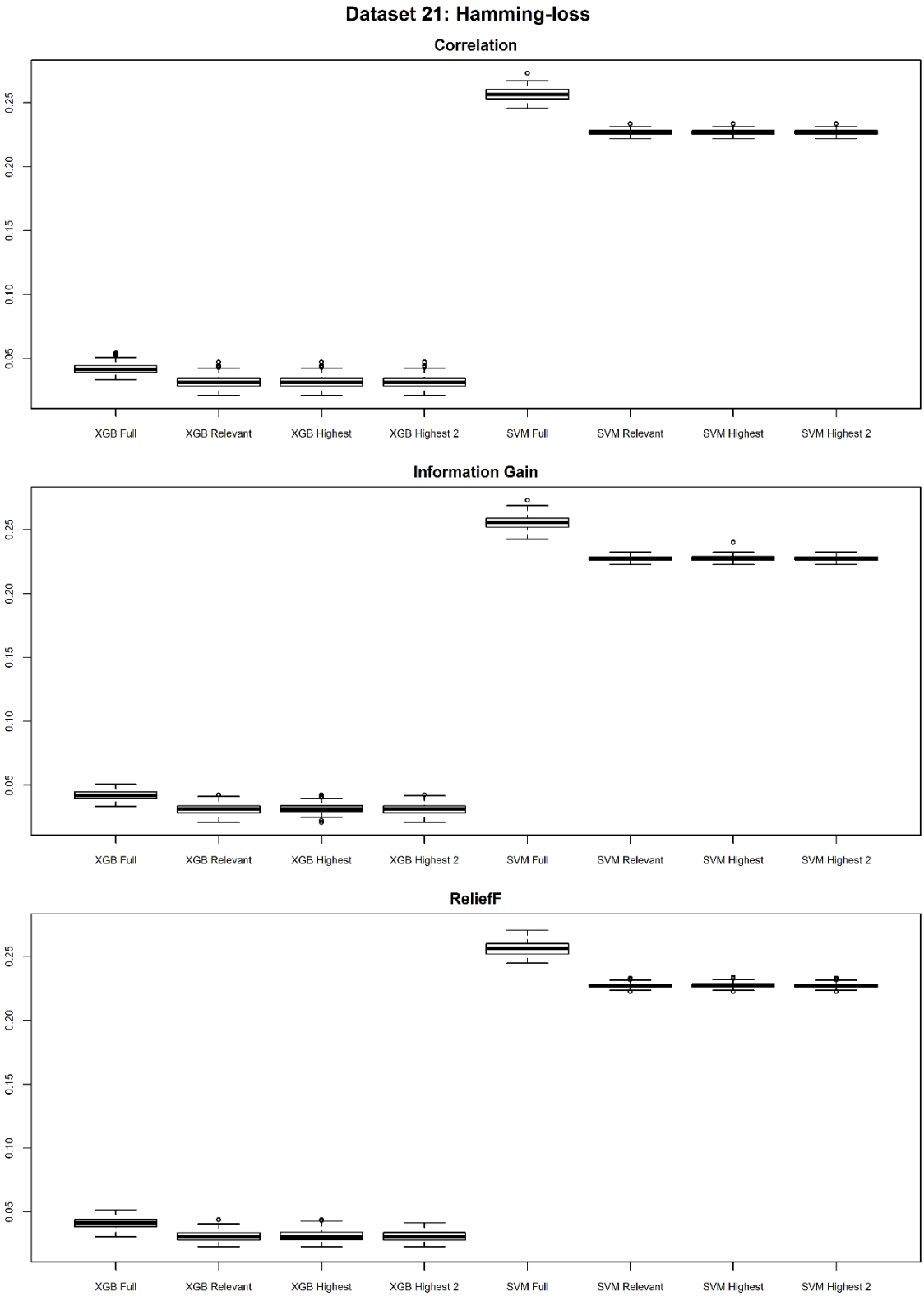


Figure F.81 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 21.

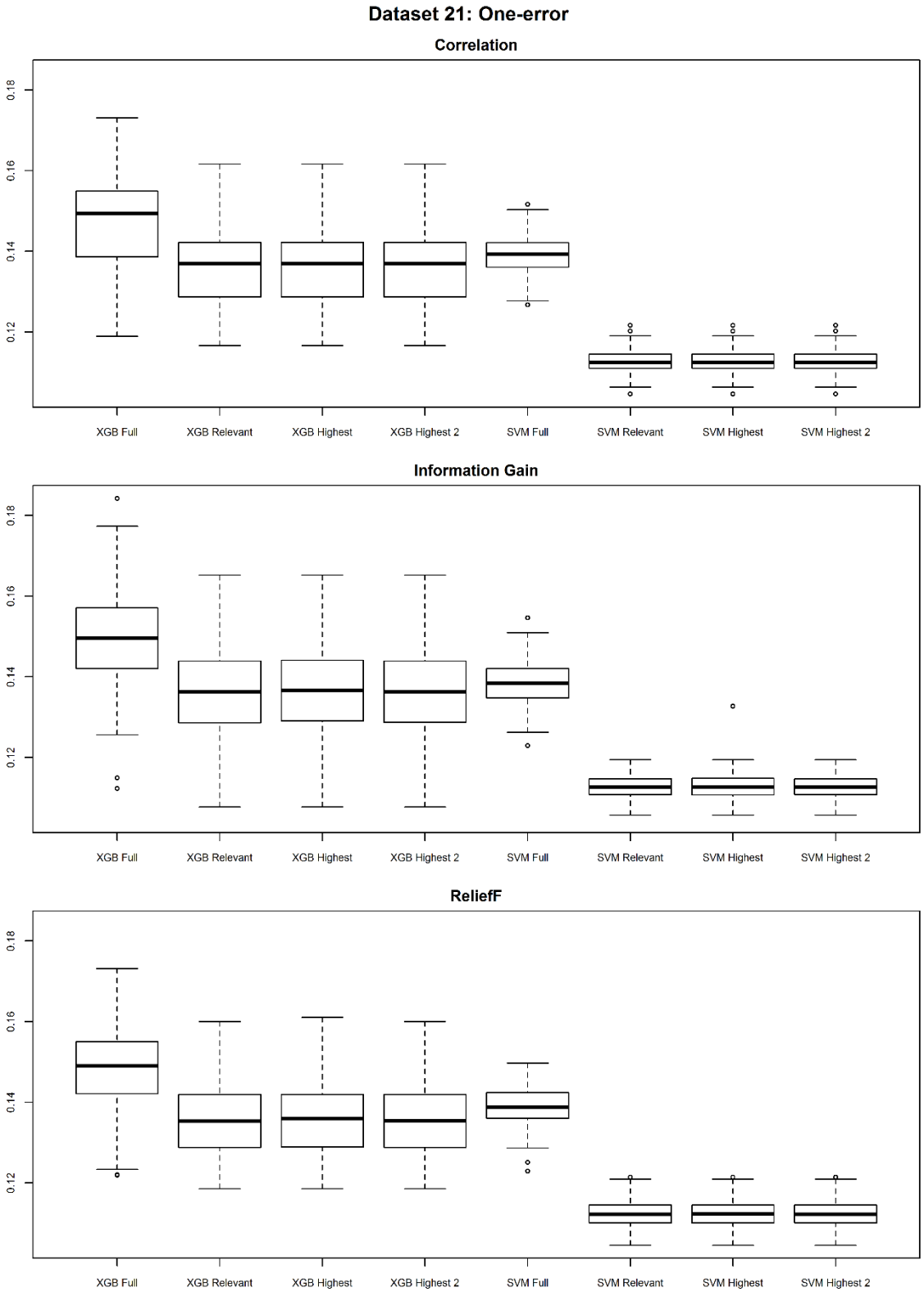


Figure F.82 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 21.

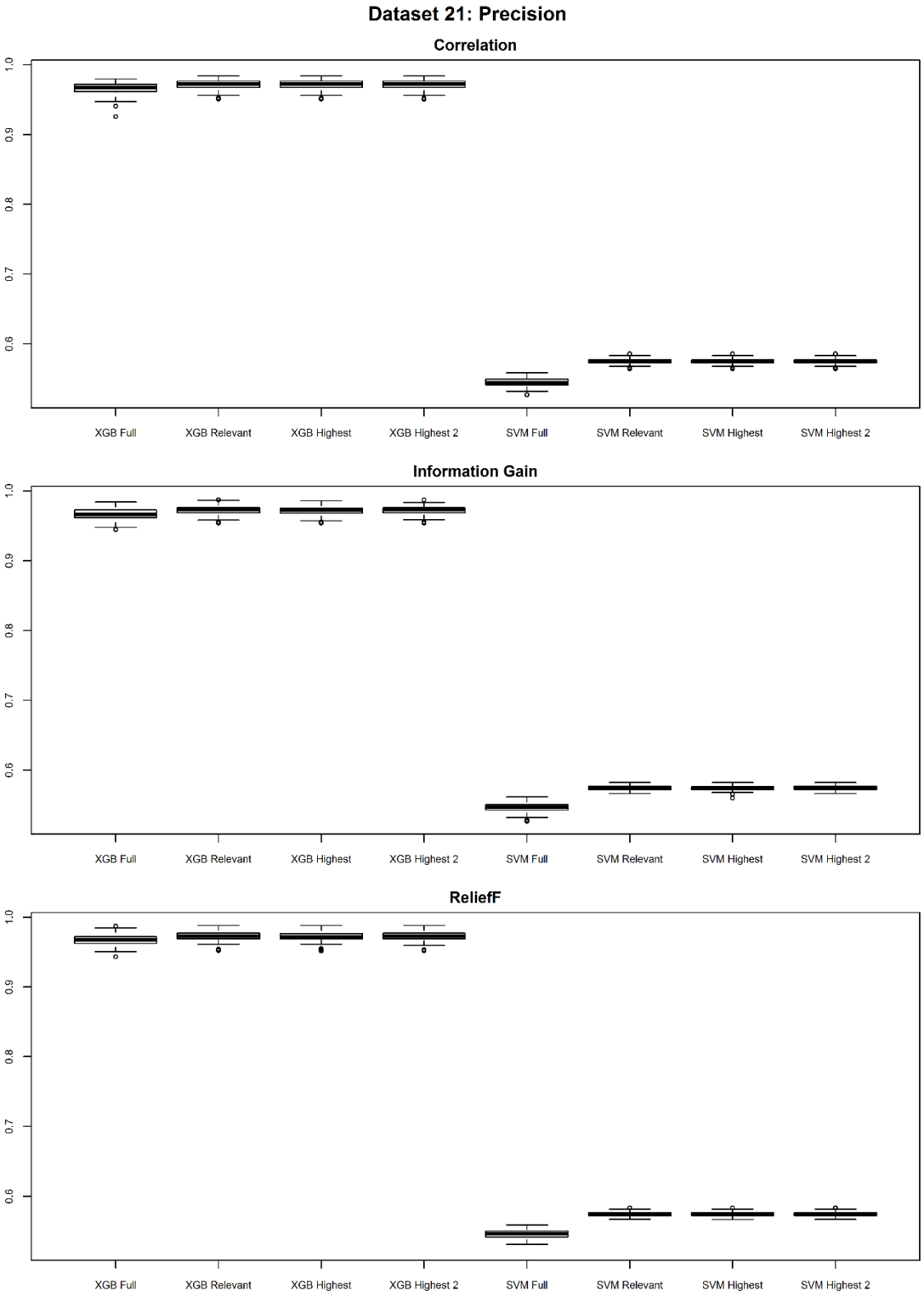


Figure F.83 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 21.

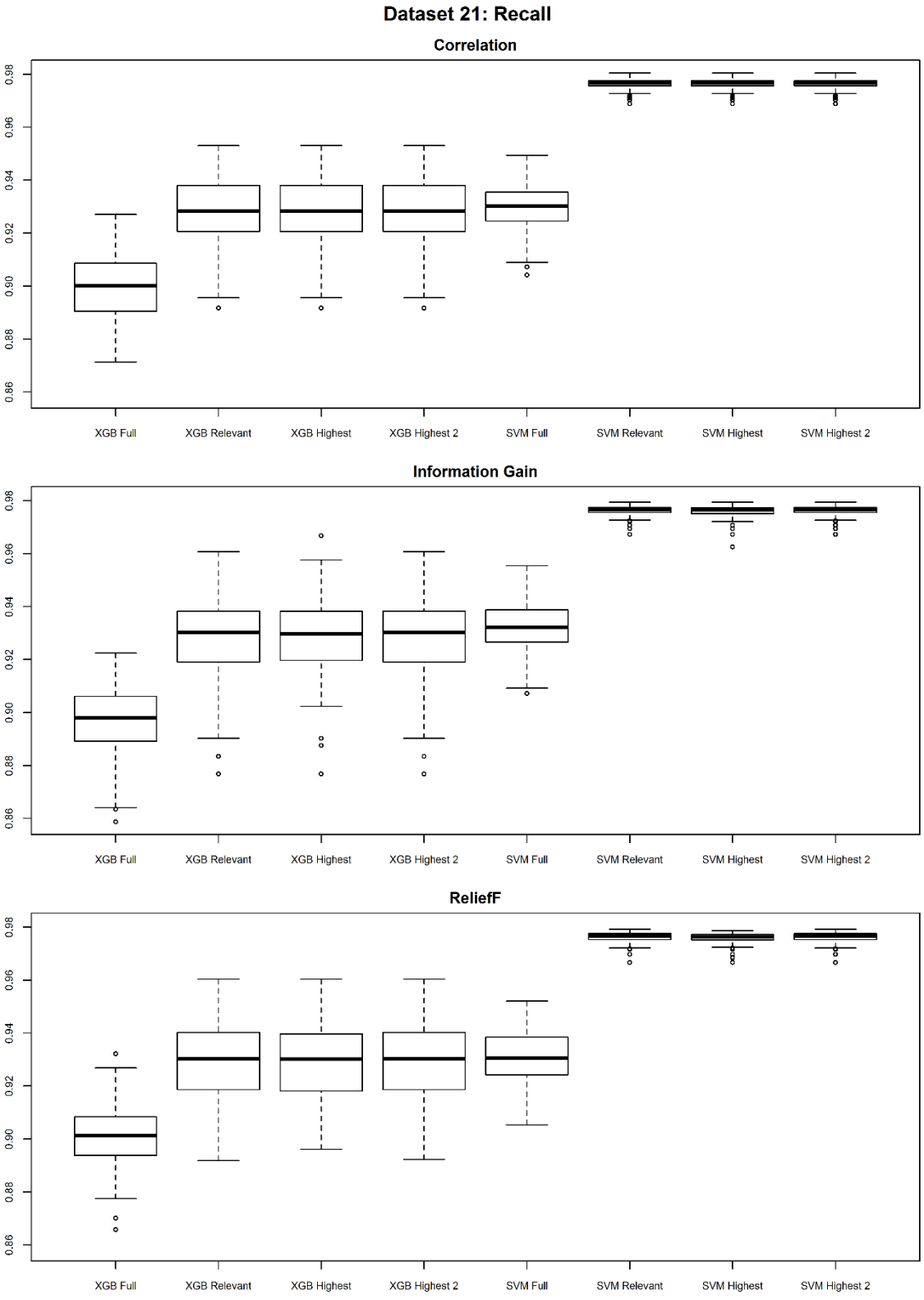


Figure F.84 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 21.

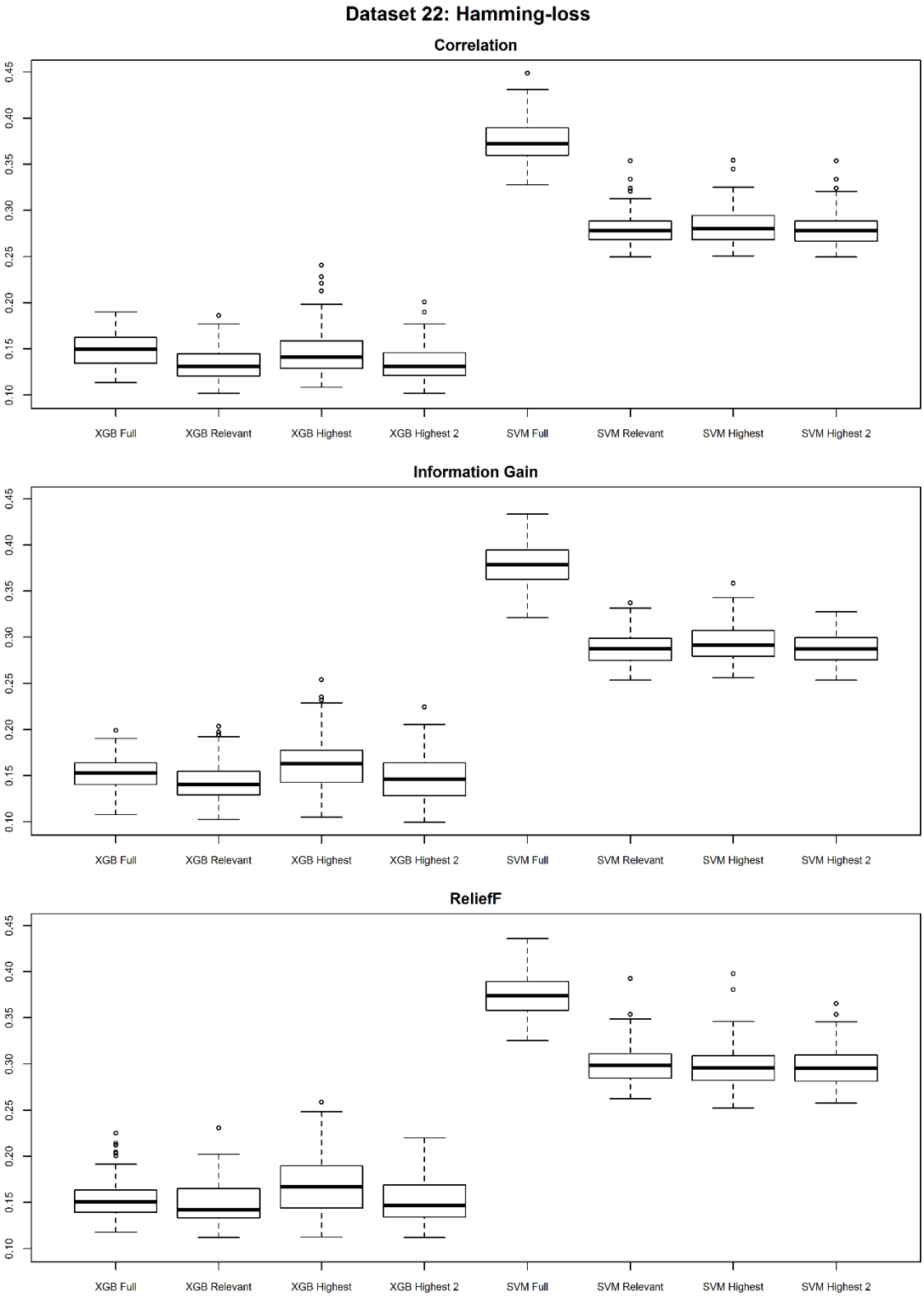


Figure F.85 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 22.

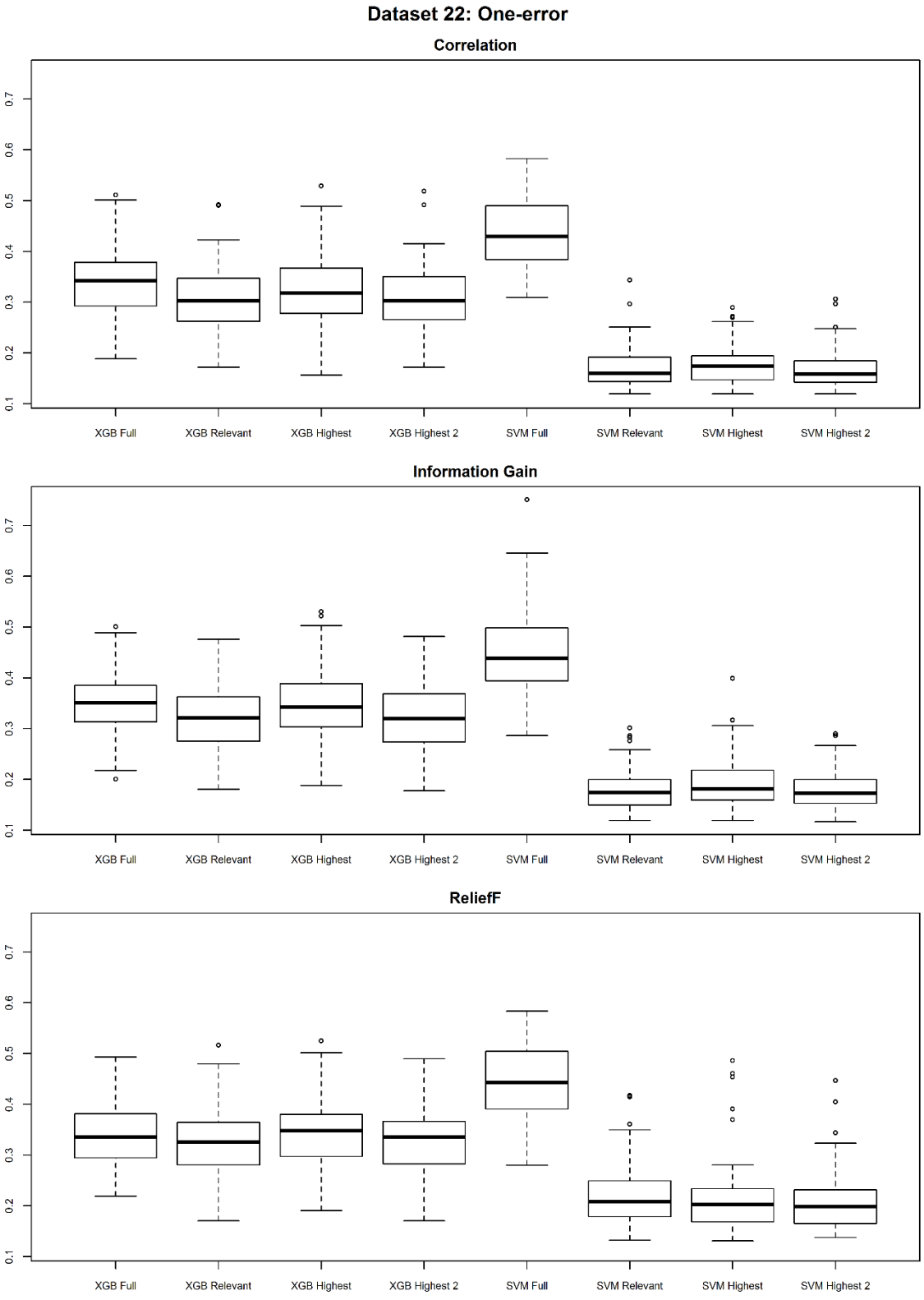


Figure F.86 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 22.

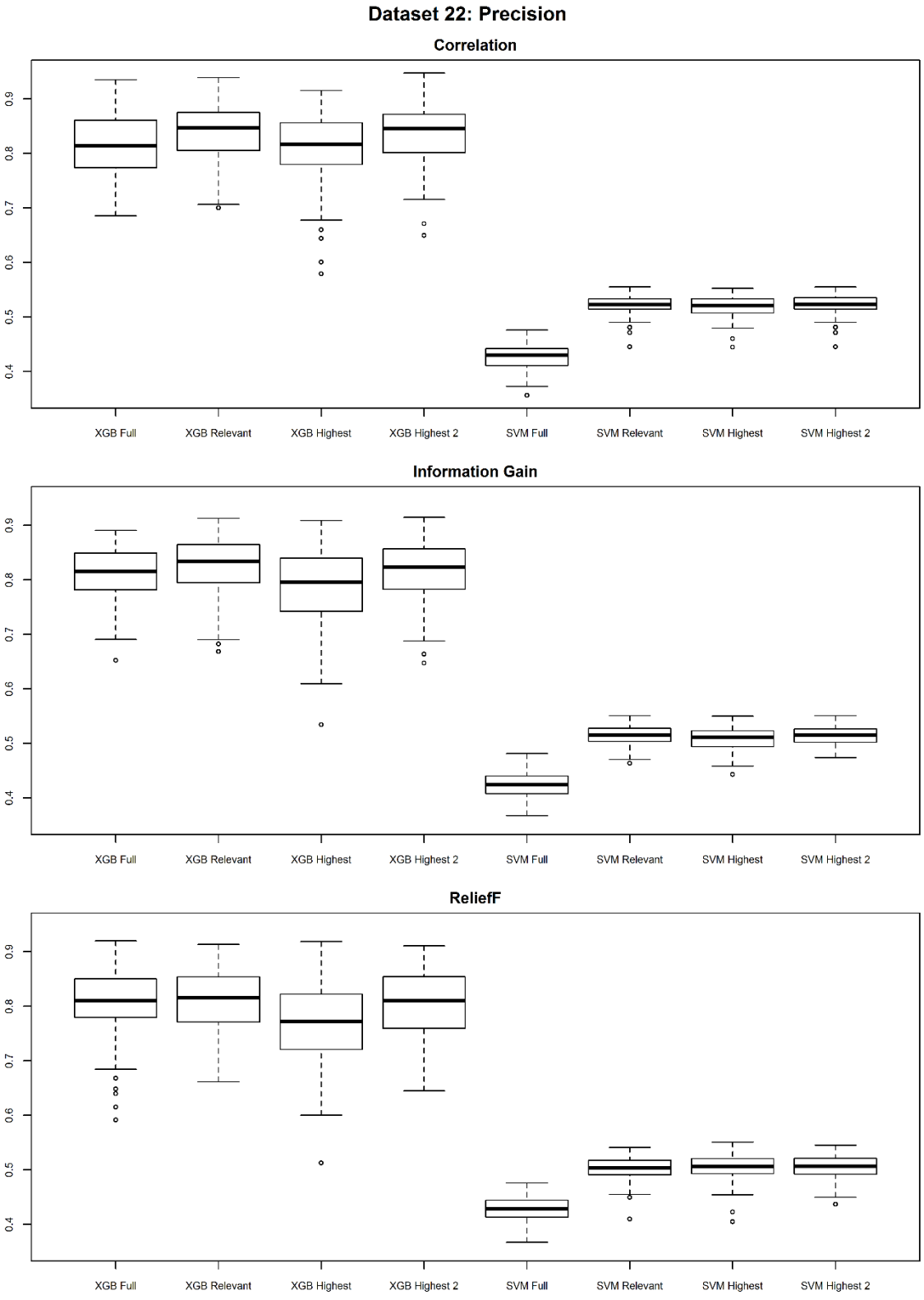


Figure F.87 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 22.

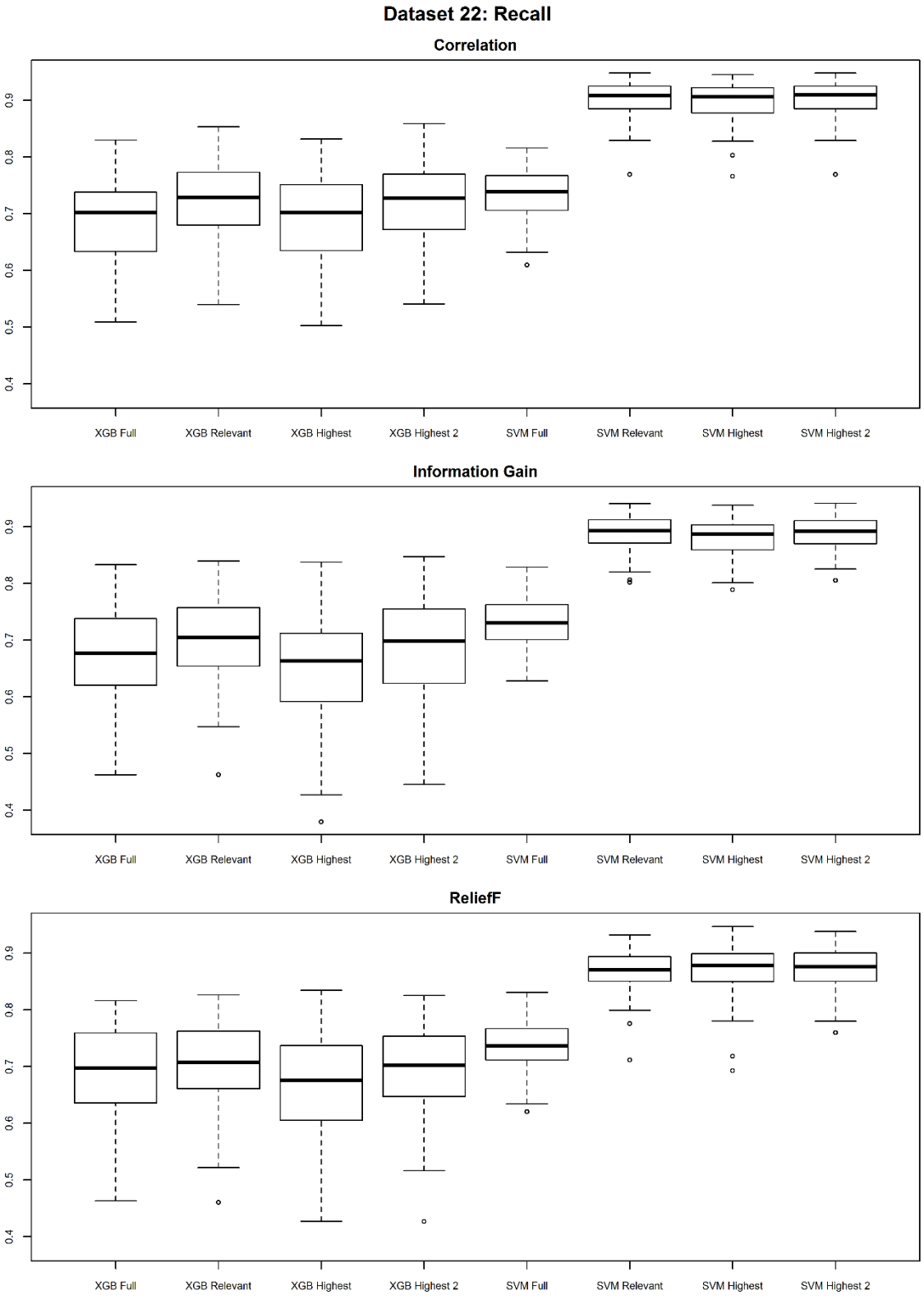


Figure F.88 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 22.

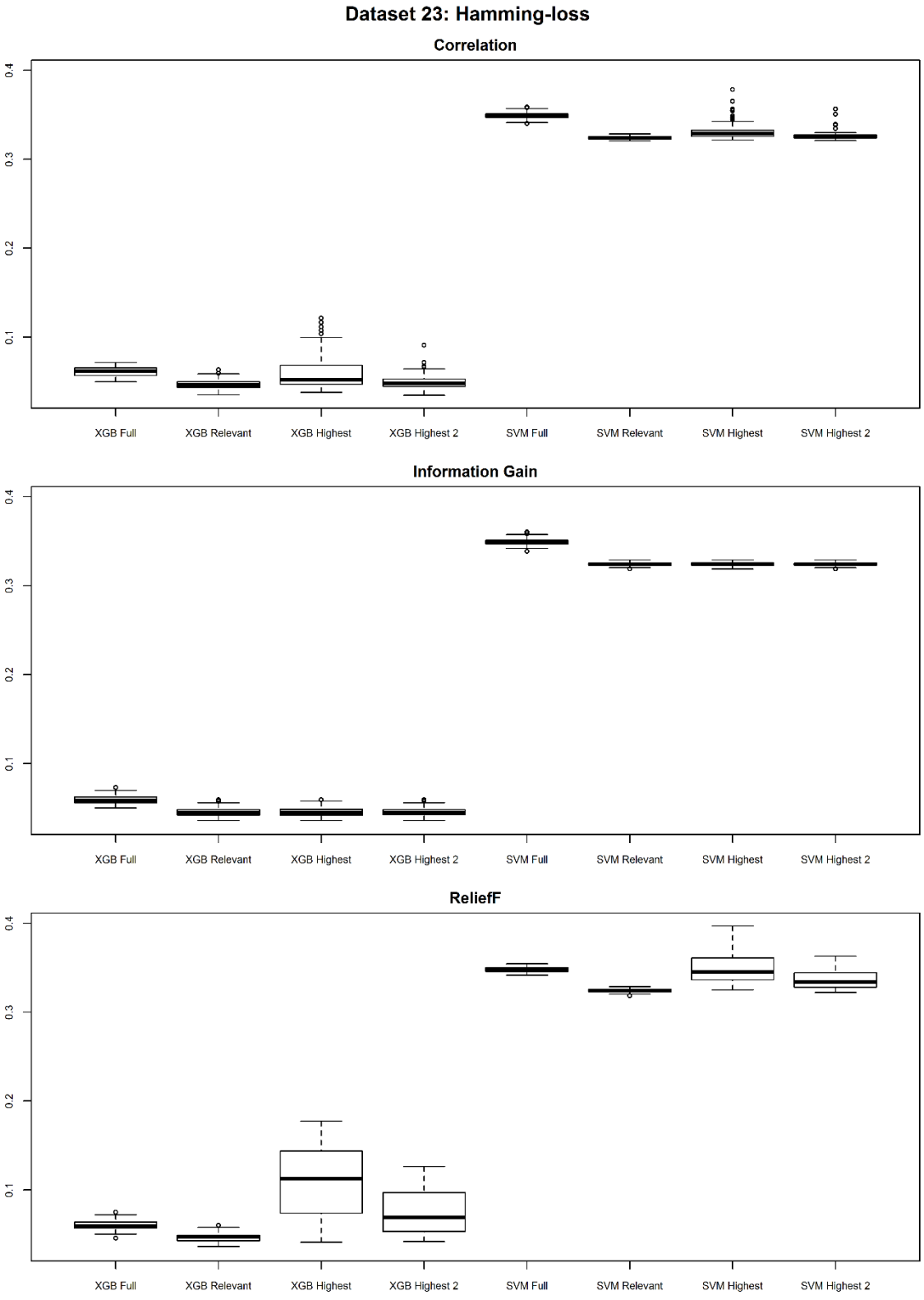


Figure F.89 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 23.

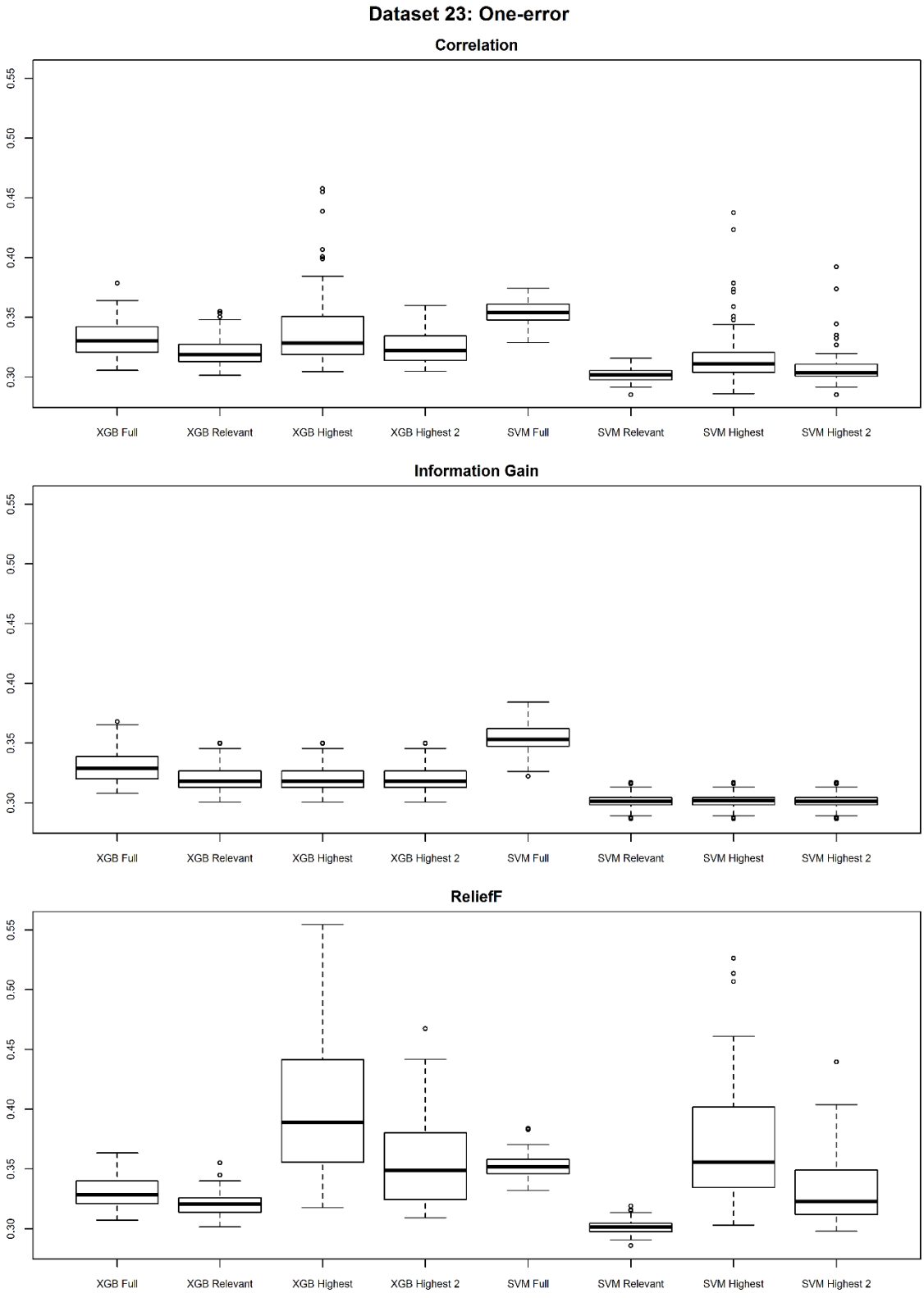


Figure F.90 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 23.

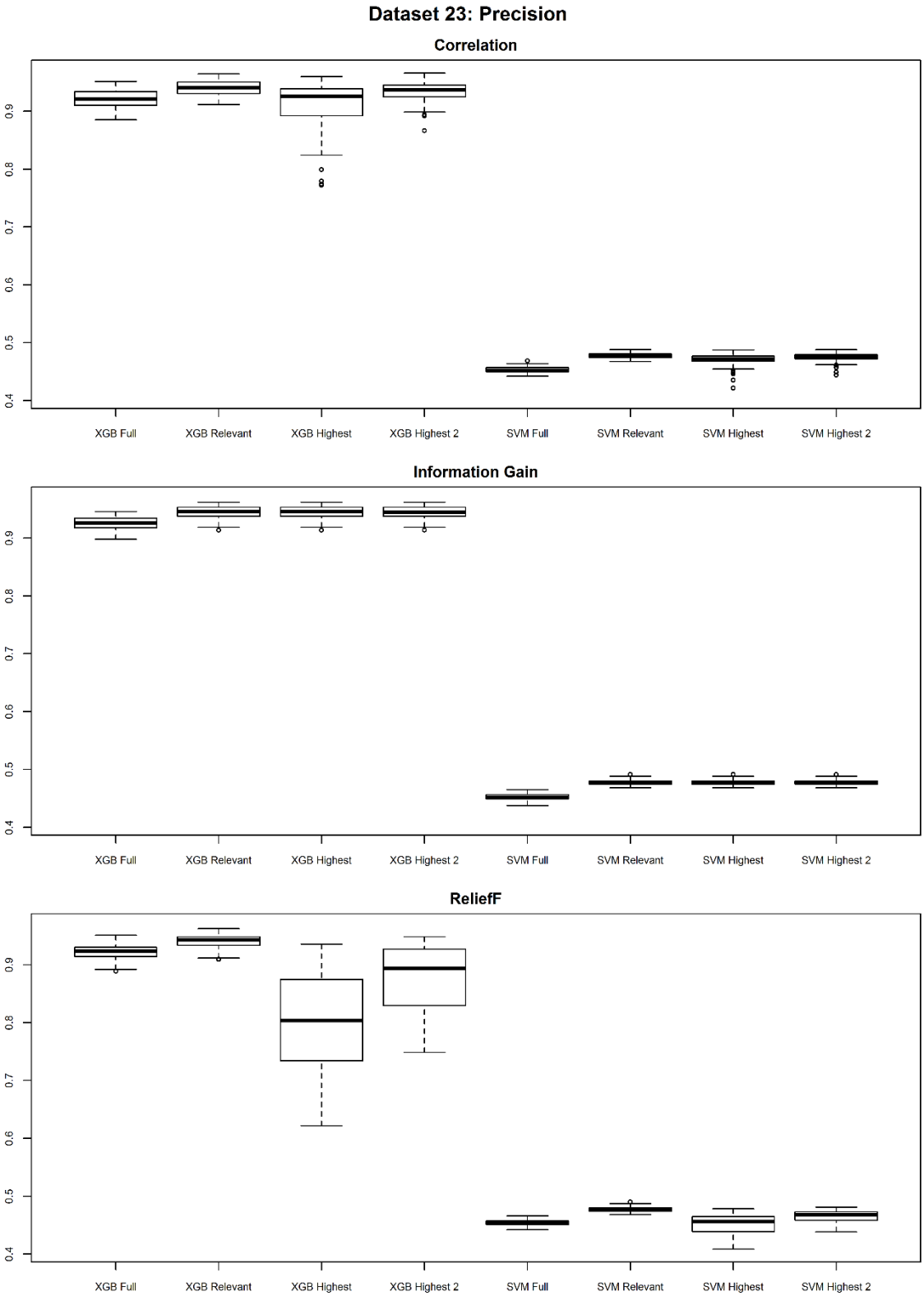


Figure F.91 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 23.

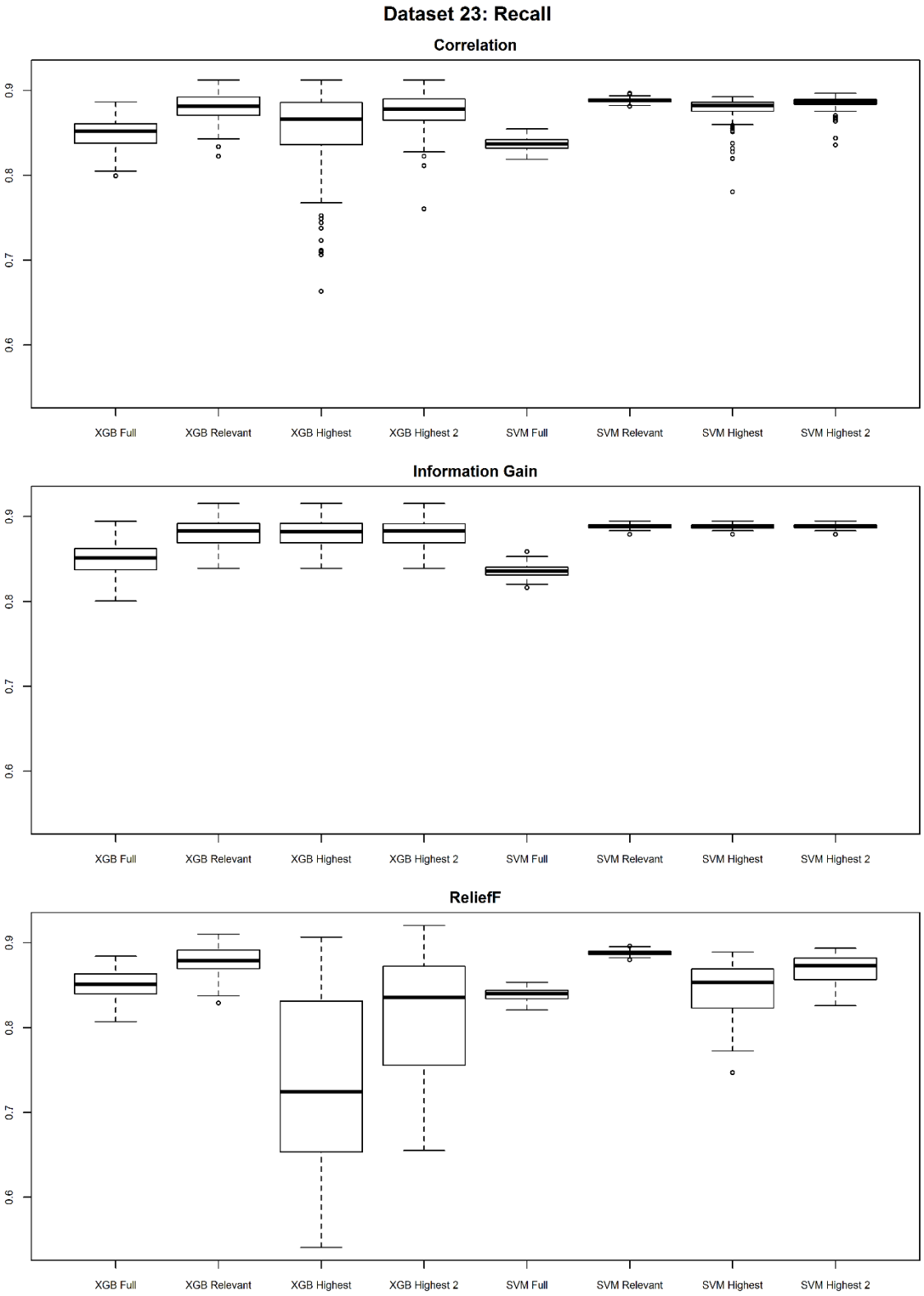


Figure F.92 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 23.

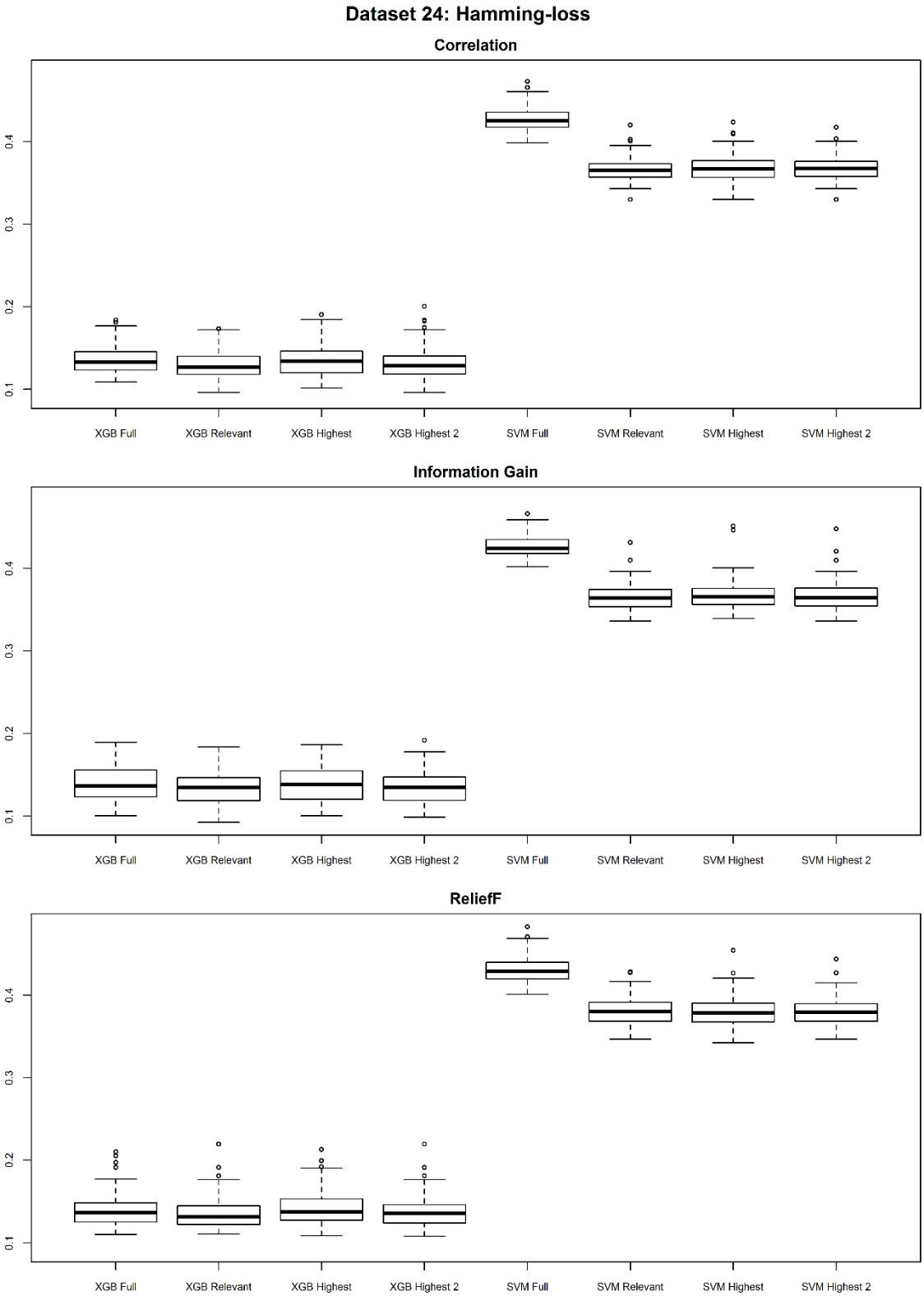


Figure F.93 Comparing SVM and XGBoost classifiers with respect to Hamming-loss: Dataset 24.

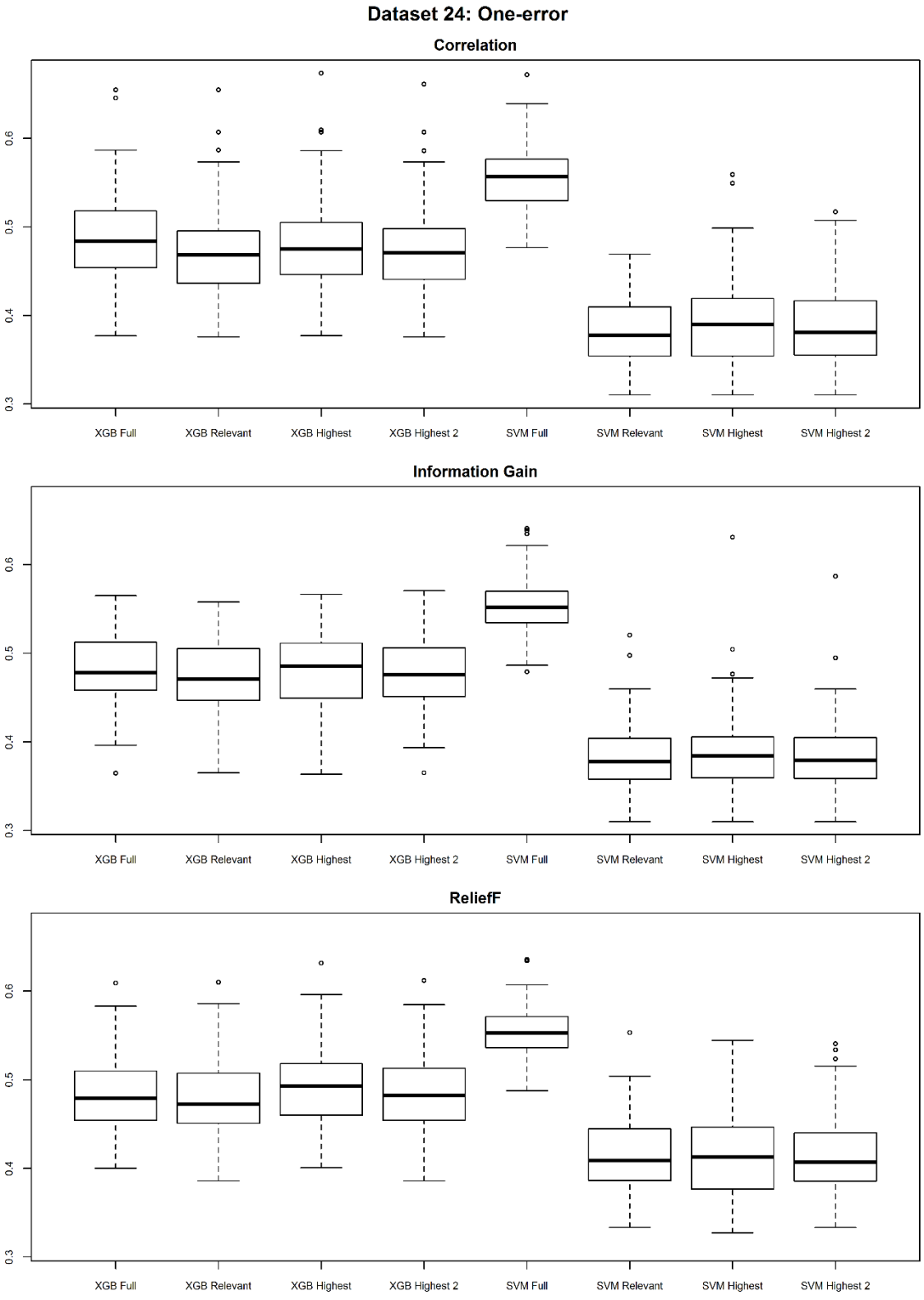


Figure F.94 Comparing SVM and XGBoost classifiers with respect to One-error: Dataset 24.

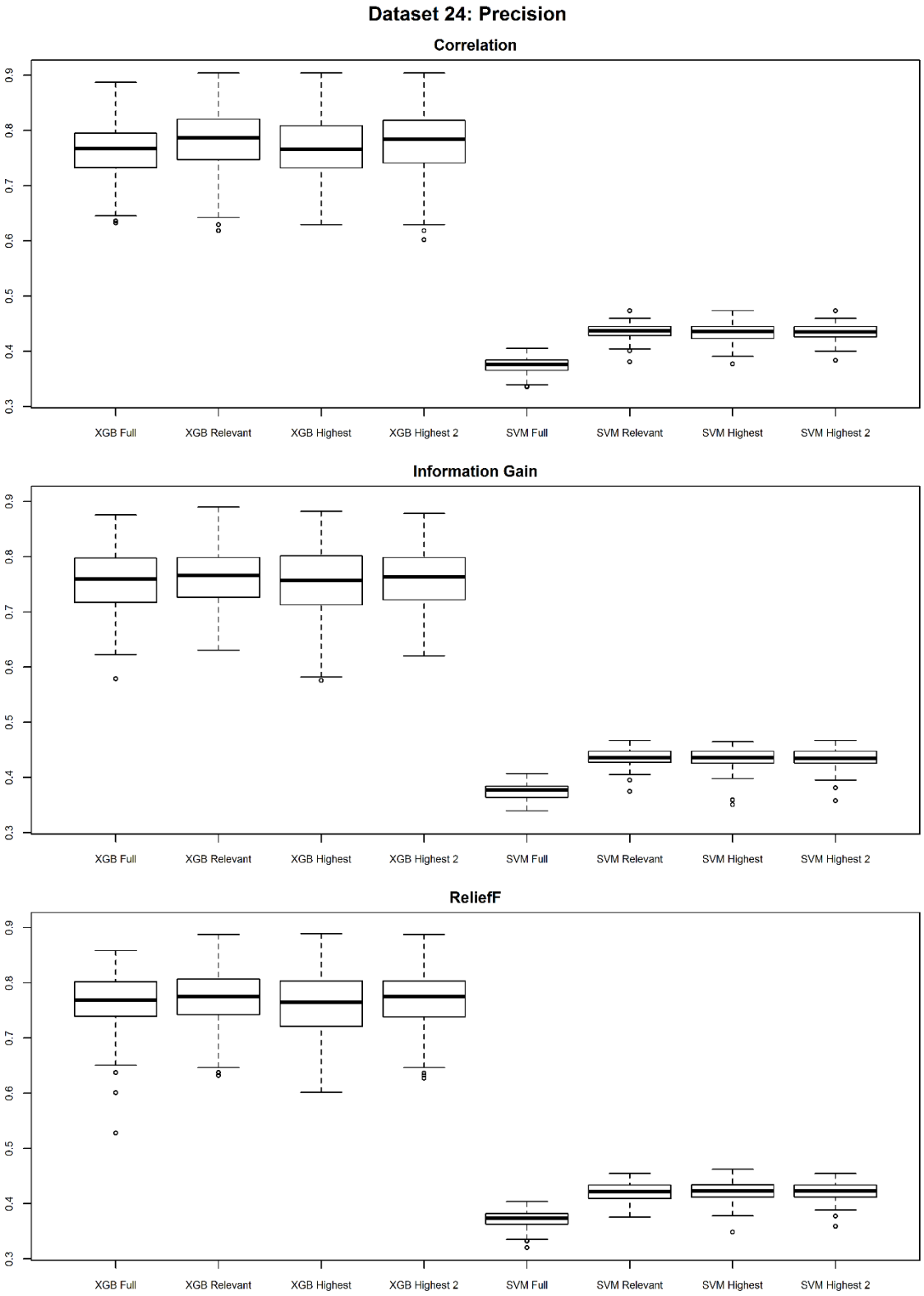


Figure F.95 Comparing SVM and XGBoost classifiers with respect to Precision: Dataset 24.

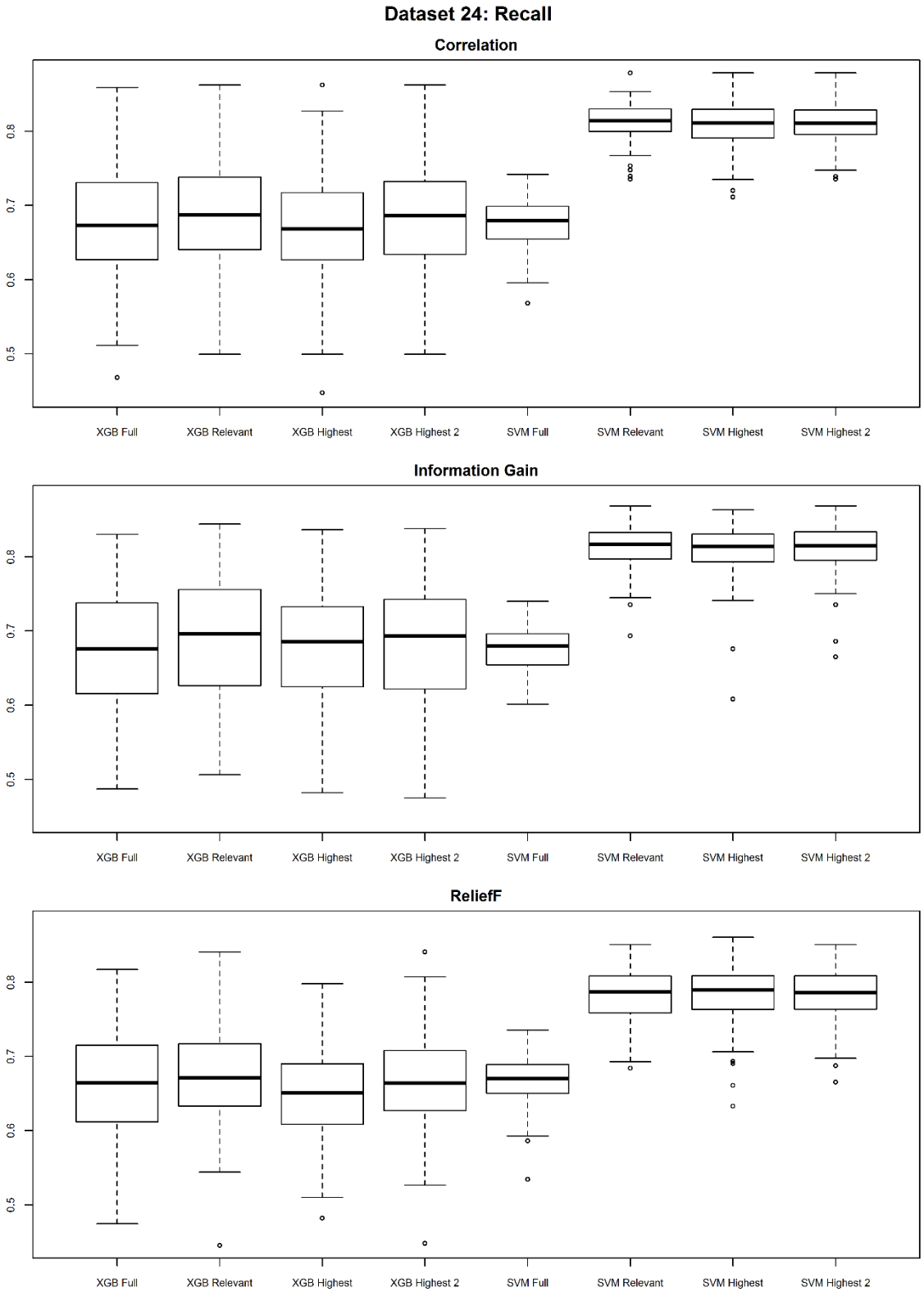


Figure F.96 Comparing SVM and XGBoost classifiers with respect to Recall: Dataset 24.

APPENDIX G

Table G.1: Three-way ANOVA for Dataset 1.

	Hamming-loss				
Dataset 1	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.003035	0.001518	6.655292	0.001311
Model	3	0.201511	0.06717	294.5511	1.5E-162
Technique	1	0.461302	0.461302	2022.877	0
Measure:Model	6	0.004686	0.000781	3.425003	0.002275
Measure:Technique	2	0.000749	0.000375	1.643097	0.1936
Model:Technique	3	0.005695	0.001898	8.323803	1.66E-05
Measure:Model:Technique	6	0.000427	7.11E-05	0.311715	0.931161
Residuals	2376	0.541829	0.000228		

	One-error				
Dataset 1	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.018018	0.009009	12.3402	4.66E-06
Model	3	0.615183	0.205061	280.8861	4.9E-156
Technique	1	3.367197	3.367197	4612.279	0
Measure:Model	6	0.017155	0.002859	3.916429	0.000672
Measure:Technique	2	0.000595	0.000297	0.407504	0.665355
Model:Technique	3	0.013461	0.004487	6.146107	0.000369
Measure:Model:Technique	6	0.001676	0.000279	0.382608	0.890514
Residuals	2376	1.7346	0.00073		

	Precision				
Dataset 1	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.007909	0.003954	7.680454	0.000473
Model	3	0.26729	0.089097	173.0467	1.8E-101
Technique	1	3.579631	3.579631	6952.496	0
Measure:Model	6	0.008251	0.001375	2.67081	0.013837
Measure:Technique	2	0.004347	0.002174	4.221678	0.014784
Model:Technique	3	0.006606	0.002202	4.276545	0.005095
Measure:Model:Technique	6	0.001463	0.000244	0.473499	0.828438
Residuals	2376	1.223331	0.000515		

	Recall				
Dataset 1	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.008747	0.004374	5.484845	0.004202
Model	3	0.497223	0.165741	207.857	1E-119
Technique	1	23.00585	23.00585	28851.81	0
Measure:Model	6	0.009122	0.00152	1.906631	0.076179
Measure:Technique	2	0.000926	0.000463	0.580387	0.559761
Model:Technique	3	0.008091	0.002697	3.38247	0.017509
Measure:Model:Technique	6	0.000634	0.000106	0.132484	0.992195
Residuals	2376	1.894575	0.000797		

Table G.2: Three-way ANOVA for Dataset 2.

Hamming-loss

Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006871	0.003436	26.90771	2.78E-12
Model	3	0.103941	0.034647	271.3538	1.9E-151
Technique	1	21.27055	21.27055	166590	0
Measure:Model	6	0.005975	0.000996	7.799672	2.48E-08
Measure:Technique	2	0.001424	0.000712	5.57799	0.00383
Model:Technique	3	0.023726	0.007909	61.94096	1.49E-38
Measure:Model:Technique	6	0.000107	1.78E-05	0.139415	0.991043
Residuals	2376	0.303372	0.000128		

One-error

Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.012806	0.006403	14.03974	8.68E-07
Model	3	0.415884	0.138628	303.9743	5.2E-167
Technique	1	2.037702	2.037702	4468.134	0
Measure:Model	6	0.015943	0.002657	5.826375	4.85E-06
Measure:Technique	2	0.010556	0.005278	11.57306	9.96E-06
Model:Technique	3	0.04771	0.015903	34.87169	4.74E-22
Measure:Model:Technique	6	0.002172	0.000362	0.793686	0.574766
Residuals	2376	1.08358	0.000456		

Precision

Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.015084	0.007542	13.05202	2.3E-06
Model	3	0.159157	0.053052	91.81374	3.23E-56
Technique	1	15.4427	15.4427	26725.59	0
Measure:Model	6	0.012469	0.002078	3.596424	0.001492
Measure:Technique	2	0.000179	8.97E-05	0.155234	0.856224
Model:Technique	3	0.022279	0.007426	12.85195	2.5E-08
Measure:Model:Technique	6	0.00106	0.000177	0.305661	0.934247
Residuals	2376	1.372911	0.000578		

Recall

Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.007419	0.00371	3.850328	0.021406
Model	3	0.397466	0.132489	137.5115	3.76E-82
Technique	1	7.906027	7.906027	8205.762	0
Measure:Model	6	0.014311	0.002385	2.475673	0.021702
Measure:Technique	2	0.025548	0.012774	13.25844	1.88E-06
Model:Technique	3	0.05196	0.01732	17.97655	1.55E-11
Measure:Model:Technique	6	0.001452	0.000242	0.251246	0.958944
Residuals	2376	2.289211	0.000963		

Table G.3: Three-way ANOVA for Dataset 3.

Hamming-loss

Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.003131	0.001566	14.48334	5.6E-07
Model	3	0.124214	0.041405	383.0096	6.7E-203
Technique	1	6.076511	6.076511	56210.09	0
Measure:Model	6	0.001739	0.00029	2.681432	0.013499
Measure:Technique	2	9.58E-05	4.79E-05	0.443296	0.64197
Model:Technique	3	0.019278	0.006426	59.44297	4.73E-37
Measure:Model:Technique	6	5.96E-05	9.93E-06	0.091833	0.997156
Residuals	2376	0.256854	0.000108		

One-error

Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006287	0.003144	8.352794	0.000243
Model	3	0.162209	0.05407	143.6683	1.5E-85
Technique	1	2.329503	2.329503	6189.708	0
Measure:Model	6	0.001449	0.000241	0.641496	0.697089
Measure:Technique	2	0.003431	0.001716	4.558737	0.010567
Model:Technique	3	0.016624	0.005541	14.72406	1.68E-09
Measure:Model:Technique	6	0.000118	1.97E-05	0.052452	0.999421
Residuals	2376	0.89421	0.000376		

Precision

Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.003163	0.001582	6.30234	0.001863
Model	3	0.118896	0.039632	157.9112	2.51E-93
Technique	1	24.52454	24.52454	97716.18	0
Measure:Model	6	0.001989	0.000331	1.320788	0.244158
Measure:Technique	2	0.000148	7.38E-05	0.294192	0.745161
Model:Technique	3	0.021807	0.007269	28.96251	2.18E-18
Measure:Model:Technique	6	0.000188	3.14E-05	0.125174	0.99331
Residuals	2376	0.596322	0.000251		

Recall

Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010555	0.005278	11.2388	1.39E-05
Model	3	0.245839	0.081946	174.5059	3E-102
Technique	1	3.114869	3.114869	6633.155	0
Measure:Model	6	0.003156	0.000526	1.120256	0.347745
Measure:Technique	2	0.002766	0.001383	2.944726	0.052809
Model:Technique	3	0.011202	0.003734	7.951904	2.83E-05
Measure:Model:Technique	6	0.000225	3.76E-05	0.079984	0.99807
Residuals	2376	1.115748	0.00047		

Table G.4: Three-way ANOVA for Dataset 4.

Hamming-loss

Dataset 4	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.016924	0.008462	62.14241	4.94E-27
Model	3	0.098596	0.032865	241.3467	1.1E-136
Technique	1	36.10776	36.10776	265157.4	0
Measure:Model	6	0.015092	0.002515	18.47186	4.38E-21
Measure:Technique	2	0.003093	0.001546	11.35571	1.24E-05
Model:Technique	3	0.017571	0.005857	43.00988	4.63E-27
Measure:Model:Technique	6	0.0014	0.000233	1.712897	0.113972
Residuals	2376	0.323551	0.000136		

One-error

Dataset 4	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.019931	0.009966	14.1691	7.64E-07
Model	3	0.486765	0.162255	230.6941	2.4E-131
Technique	1	2.409739	2.409739	3426.164	0
Measure:Model	6	0.037229	0.006205	8.821963	1.55E-09
Measure:Technique	2	0.006153	0.003076	4.374127	0.012701
Model:Technique	3	0.126563	0.042188	59.98251	2.24E-37
Measure:Model:Technique	6	0.000869	0.000145	0.205849	0.975079
Residuals	2376	1.671123	0.000703		

Precision

Dataset 4	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.033402	0.016701	26.83081	3E-12
Model	3	0.146197	0.048732	78.29117	2.71E-48
Technique	1	54.43828	54.43828	87458.02	0
Measure:Model	6	0.041218	0.00687	11.03656	3.63E-12
Measure:Technique	2	0.013297	0.006649	10.68117	2.41E-05
Model:Technique	3	0.02494	0.008313	13.35596	1.21E-08
Measure:Model:Technique	6	0.013004	0.002167	3.482015	0.001978
Residuals	2376	1.478942	0.000622		

Recall

Dataset 4	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.066837	0.033418	37.21597	1.22E-16
Model	3	0.364727	0.121576	135.3913	5.63E-81
Technique	1	0.337191	0.337191	375.5089	8.72E-78
Measure:Model	6	0.047789	0.007965	8.869979	1.36E-09
Measure:Technique	2	0.017582	0.008791	9.789944	5.83E-05
Model:Technique	3	0.026241	0.008747	9.740943	2.19E-06
Measure:Model:Technique	6	0.005547	0.000925	1.029588	0.403907
Residuals	2376	2.133547	0.000898		

Table G.5: Three-way ANOVA for Dataset 5.

Hamming-loss

Dataset 5	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.024703	0.012351	106.2429	6.41E-45
Model	3	0.100762	0.033587	288.9104	7E-160
Technique	1	5.518671	5.518671	47470.2	0
Measure:Model	6	0.007974	0.001329	11.43193	1.22E-12
Measure:Technique	2	0.00081	0.000405	3.485525	0.030794
Model:Technique	3	0.00337	0.001123	9.662196	2.45E-06
Measure:Model:Technique	6	0.000187	3.12E-05	0.268093	0.951902
Residuals	2376	0.276223	0.000116		

One-error

Dataset 5	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.125691	0.062846	102.1771	2.69E-43
Model	3	0.55789	0.185963	302.3471	3E-166
Technique	1	5.062846	5.062846	8231.381	0
Measure:Model	6	0.03341	0.005568	9.053331	8.27E-10
Measure:Technique	2	0.001257	0.000628	1.021819	0.360098
Model:Technique	3	0.028646	0.009549	15.52436	5.32E-10
Measure:Model:Technique	6	0.000502	8.36E-05	0.135906	0.991638
Residuals	2376	1.461398	0.000615		

Precision

Dataset 5	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.037761	0.01888	29.96666	1.4E-13
Model	3	0.149089	0.049696	78.87701	1.22E-48
Technique	1	24.63148	24.63148	39094.53	0
Measure:Model	6	0.020735	0.003456	5.485145	1.19E-05
Measure:Technique	2	0.000274	0.000137	0.217805	0.804298
Model:Technique	3	0.001993	0.000664	1.05432	0.367386
Measure:Model:Technique	6	0.003684	0.000614	0.974643	0.440708
Residuals	2376	1.496997	0.00063		

Recall

Dataset 5	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.109991	0.054995	64.21101	6.93E-28
Model	3	0.504684	0.168228	196.4182	8.6E-114
Technique	1	55.13672	55.13672	64375.99	0
Measure:Model	6	0.024718	0.00412	4.809955	6.9E-05
Measure:Technique	2	0.008921	0.004461	5.208031	0.005535
Model:Technique	3	0.005074	0.001691	1.974829	0.115651
Measure:Model:Technique	6	0.000485	8.08E-05	0.094312	0.996936
Residuals	2376	2.034995	0.000856		

Table G.6: Three-way ANOVA for Dataset 6.

Hamming-loss

Dataset 6	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004279	0.00214	18.92963	6.98E-09
Model	3	0.066368	0.022123	195.7152	2E-113
Technique	1	21.79393	21.79393	192807.8	0
Measure:Model	6	0.002687	0.000448	3.962432	0.000599
Measure:Technique	2	0.001433	0.000716	6.336794	0.0018
Model:Technique	3	0.004165	0.001388	12.28322	5.66E-08
Measure:Model:Technique	6	0.000704	0.000117	1.038332	0.39824
Residuals	2376	0.26857	0.000113		

One-error

Dataset 6	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010815	0.005407	9.462213	8.07E-05
Model	3	0.332872	0.110957	194.1573	1.3E-112
Technique	1	3.711061	3.711061	6493.746	0
Measure:Model	6	0.007104	0.001184	2.071691	0.053458
Measure:Technique	2	0.005985	0.002992	5.236195	0.005382
Model:Technique	3	0.036997	0.012332	21.57963	8.68E-14
Measure:Model:Technique	6	0.002526	0.000421	0.736744	0.620024
Residuals	2376	1.357842	0.000571		

Precision

Dataset 6	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.011965	0.005982	7.616849	0.000504
Model	3	0.144738	0.048246	61.42875	3.02E-38
Technique	1	27.61552	27.61552	35161.12	0
Measure:Model	6	0.010068	0.001678	2.1364	0.046414
Measure:Technique	2	0.011438	0.005719	7.281522	0.000704
Model:Technique	3	0.016932	0.005644	7.186238	8.42E-05
Measure:Model:Technique	6	0.004913	0.000819	1.042488	0.395565
Residuals	2376	1.866109	0.000785		

Recall

Dataset 6	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.016306	0.008153	7.394029	0.000629
Model	3	0.331533	0.110511	100.2253	4.35E-61
Technique	1	37.53847	37.53847	34044.59	0
Measure:Model	6	0.006329	0.001055	0.956698	0.453159
Measure:Technique	2	0.004314	0.002157	1.956192	0.141623
Model:Technique	3	0.016882	0.005627	5.103694	0.001604
Measure:Model:Technique	6	0.00258	0.00043	0.389974	0.885857
Residuals	2376	2.619841	0.001103		

Table G.7: Three-way ANOVA for Dataset 7.

Hamming-loss

Dataset 7	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.934605	0.467302	2953.321	0
Model	3	0.028136	0.009379	59.2722	6E-37
Technique	1	23.96428	23.96428	151452.7	0
Measure:Model	6	0.047416	0.007903	49.94421	4.68E-58
Measure:Technique	2	0.895203	0.447601	2828.812	0
Model:Technique	3	0.00412	0.001373	8.680184	9.99E-06
Measure:Model:Technique	6	0.042238	0.00704	44.49027	8.52E-52
Residuals	2376	0.375953	0.000158		

One-error

Dataset 7	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	4.894901	2.447451	2911.578	0
Model	3	0.062057	0.020686	24.60853	1.12E-15
Technique	1	0.0139	0.0139	16.53646	4.93E-05
Measure:Model	6	0.176803	0.029467	35.05526	8.22E-41
Measure:Technique	2	4.568802	2.284401	2717.608	0
Model:Technique	3	0.051123	0.017041	20.2727	5.69E-13
Measure:Model:Technique	6	0.178595	0.029766	35.41045	3.15E-41
Residuals	2376	1.997247	0.000841		

Precision

Dataset 7	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	1.03243	0.516215	1540.876	0
Model	3	0.02574	0.00858	25.61048	2.65E-16
Technique	1	95.07671	95.07671	283799.4	0
Measure:Model	6	0.049335	0.008222	24.54366	2.33E-28
Measure:Technique	2	0.83407	0.417035	1244.83	0
Model:Technique	3	0.003653	0.001218	3.634783	0.012383
Measure:Model:Technique	6	0.041601	0.006933	20.69607	9.38E-24
Residuals	2376	0.795993	0.000335		

Recall

Dataset 7	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	2.219538	1.109769	1411.211	0
Model	3	0.112881	0.037627	47.84767	5.08E-30
Technique	1	2.261588	2.261588	2875.893	0
Measure:Model	6	0.138576	0.023096	29.36942	4.22E-34
Measure:Technique	2	2.422847	1.211423	1540.477	0
Model:Technique	3	0.016658	0.005553	7.060875	0.000101
Measure:Model:Technique	6	0.11466	0.01911	24.30086	4.55E-28
Residuals	2376	1.868474	0.000786		

Table G.8: Three-way ANOVA for Dataset 8.

Hamming-loss

Dataset 8	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006698	0.003349	35.26731	8.06E-16
Model	3	0.054092	0.018031	189.8748	2.3E-110
Technique	1	37.75131	37.75131	397547.4	0
Measure:Model	6	0.006633	0.001106	11.6419	6.87E-13
Measure:Technique	2	0.000123	6.17E-05	0.649275	0.522517
Model:Technique	3	0.011296	0.003765	39.65164	5.34E-25
Measure:Model:Technique	6	0.000164	2.74E-05	0.288178	0.94279
Residuals	2376	0.225626	9.5E-05		

One-error

Dataset 8	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.023693	0.011847	17.93077	1.87E-08
Model	3	0.233945	0.077982	118.0307	2.98E-71
Technique	1	2.443259	2.443259	3698.042	0
Measure:Model	6	0.024198	0.004033	6.104227	2.33E-06
Measure:Technique	2	0.004304	0.002152	3.257431	0.038659
Model:Technique	3	0.066168	0.022056	33.38351	3.95E-21
Measure:Model:Technique	6	0.000477	7.94E-05	0.120229	0.994007
Residuals	2376	1.569799	0.000661		

Precision

Dataset 8	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.017264	0.008632	15.09998	3.04E-07
Model	3	0.085167	0.028389	49.66073	3.99E-31
Technique	1	97.83203	97.83203	171136.9	0
Measure:Model	6	0.025798	0.0043	7.521504	5.26E-08
Measure:Technique	2	0.001495	0.000748	1.307639	0.270653
Model:Technique	3	0.019273	0.006424	11.23809	2.55E-07
Measure:Model:Technique	6	0.008369	0.001395	2.439939	0.023544
Residuals	2376	1.358263	0.000572		

Recall

Dataset 8	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.049952	0.024976	22.70227	1.71E-10
Model	3	0.229128	0.076376	69.42257	4.94E-43
Technique	1	3.763855	3.763855	3421.188	0
Measure:Model	6	0.021396	0.003566	3.241321	0.00356
Measure:Technique	2	0.018436	0.009218	8.378698	0.000237
Model:Technique	3	0.043369	0.014456	13.14007	1.65E-08
Measure:Model:Technique	6	0.000913	0.000152	0.13828	0.991238
Residuals	2376	2.613981	0.0011		

Table G.9: Three-way ANOVA for Dataset 9.

Hamming-loss

Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004874	0.002437	50.65595	2.86E-22
Model	3	0.71039	0.236797	4922.316	0
Technique	1	1.63716	1.63716	34031.78	0
Measure:Model	6	0.008718	0.001453	30.20368	4.33E-35
Measure:Technique	2	0.000108	5.41E-05	1.124466	0.324999
Model:Technique	3	0.017143	0.005714	118.7859	1.12E-71
Measure:Model:Technique	6	0.000221	3.68E-05	0.764079	0.598174
Residuals	2376	0.114302	4.81E-05		

One-error

Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010785	0.005392	43.34822	3.23E-19
Model	3	1.722869	0.57429	4616.619	0
Technique	1	0.803787	0.803787	6461.512	0
Measure:Model	6	0.019977	0.00333	26.76565	5.23E-31
Measure:Technique	2	0.000914	0.000457	3.672868	0.025548
Model:Technique	3	0.088376	0.029459	236.8137	2E-134
Measure:Model:Technique	6	0.000555	9.24E-05	0.7431	0.614925
Residuals	2376	0.295565	0.000124		

Precision

Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006901	0.003451	27.3575	1.79E-12
Model	3	0.664984	0.221661	1757.425	0
Technique	1	9.17246	9.17246	72723.18	0
Measure:Model	6	0.014895	0.002483	19.68269	1.54E-22
Measure:Technique	2	0.000314	0.000157	1.244106	0.288386
Model:Technique	3	0.028259	0.00942	74.6826	3.69E-46
Measure:Model:Technique	6	0.001779	0.000297	2.351342	0.028775
Residuals	2376	0.299681	0.000126		

Recall

Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.009331	0.004665	23.67651	6.58E-11
Model	3	2.088307	0.696102	3532.62	0
Technique	1	14.29967	14.29967	72568.78	0
Measure:Model	6	0.014662	0.002444	12.40085	8.49E-14
Measure:Technique	2	0.000434	0.000217	1.102344	0.332262
Model:Technique	3	0.015159	0.005053	25.64316	2.53E-16
Measure:Model:Technique	6	0.000673	0.000112	0.568841	0.755445
Residuals	2376	0.468191	0.000197		

Table G.10: Three-way ANOVA for Dataset 10.

Hamming-loss

Dataset 10	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.192649	0.096325	185.4884	1.40E-75
Model	3	0.296885	0.098962	190.5663	9.9E-111
Technique	1	1.660305	1.660305	3197.184	0
Measure:Model	6	0.083579	0.01393	26.82408	4.46E-31
Measure:Technique	2	0.04021	0.020105	38.71537	2.85E-17
Model:Technique	3	0.140558	0.046853	90.22218	2.72E-55
Measure:Model:Technique	6	0.014821	0.00247	4.756857	7.92E-05
Residuals	2376	1.233862	0.000519		

One-error

Dataset 10	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.643947	0.321973	150.0104	4.45E-62
Model	3	1.210293	0.403431	187.9623	2.3E-109
Technique	1	0.309933	0.309933	144.4005	2.46E-32
Measure:Model	6	0.248927	0.041488	19.3296	4.09E-22
Measure:Technique	2	0.206749	0.103375	48.16316	3.13E-21
Model:Technique	3	0.769165	0.256388	119.4537	4.69E-72
Measure:Model:Technique	6	0.048942	0.008157	3.800442	0.000899
Residuals	2376	5.099704	0.002146		

Precision

Dataset 10	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.252974	0.126487	133.4165	1.22E-55
Model	3	0.324559	0.108186	114.1131	4.94E-69
Technique	1	1.584896	1.584896	1671.72	3.5E-277
Measure:Model	6	0.110466	0.018411	19.41956	3.19E-22
Measure:Technique	2	0.016697	0.008348	8.805596	0.000155
Model:Technique	3	0.126238	0.042079	44.38452	6.65E-28
Measure:Model:Technique	6	0.010806	0.001801	1.899746	0.077297
Residuals	2376	2.252598	0.000948		

Recall

Dataset 10	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.343385	0.171693	76.83188	4.64E-33
Model	3	0.643188	0.214396	95.94144	1.3E-58
Technique	1	15.58888	15.58888	6975.967	0
Measure:Model	6	0.187079	0.03118	13.95289	1.17E-15
Measure:Technique	2	0.190092	0.095046	42.53274	7.1E-19
Model:Technique	3	0.270196	0.090065	40.3039	2.12E-25
Measure:Model:Technique	6	0.023111	0.003852	1.723688	0.111488
Residuals	2376	5.309539	0.002235		

Table G.11: Three-way ANOVA for Dataset 11.

Hamming-loss

Dataset 11	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.026812	0.013406	151.6291	1.06E-62
Model	3	0.30118	0.100393	1135.494	0
Technique	1	23.24547	23.24547	262917.1	0
Measure:Model	6	0.02909	0.004848	54.83617	1.3E-63
Measure:Technique	2	0.00021	0.000105	1.18736	0.305207
Model:Technique	3	0.075722	0.025241	285.4821	3.1E-158
Measure:Model:Technique	6	0.000394	6.57E-05	0.742918	0.615071
Residuals	2376	0.210071	8.84E-05		

One-error

Dataset 11	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.069885	0.034943	104.7583	2.51E-44
Model	3	1.544132	0.514711	1543.109	0
Technique	1	1.453289	1.453289	4356.981	0
Measure:Model	6	0.0739	0.012317	36.92566	5.27E-43
Measure:Technique	2	0.001131	0.000565	1.694912	0.183837
Model:Technique	3	0.112127	0.037376	112.0527	7.33E-68
Measure:Model:Technique	6	0.002937	0.00049	1.467548	0.185356
Residuals	2376	0.792525	0.000334		

Precision

Dataset 11	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.053228	0.026614	88.42372	9.12E-38
Model	3	0.38102	0.127007	421.9756	1E-219
Technique	1	21.18626	21.18626	70390.58	0
Measure:Model	6	0.052242	0.008707	28.92853	1.41E-33
Measure:Technique	2	0.008672	0.004336	14.4057	6.04E-07
Model:Technique	3	0.141323	0.047108	156.5131	1.44E-92
Measure:Model:Technique	6	0.004257	0.000709	2.357265	0.028393
Residuals	2376	0.715132	0.000301		

Recall

Dataset 11	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.064018	0.032009	70.71179	1.48E-30
Model	3	1.504327	0.501442	1107.74	0
Technique	1	7.370031	7.370031	16281.2	0
Measure:Model	6	0.076624	0.012771	28.21196	9.98E-33
Measure:Technique	2	0.004878	0.002439	5.388303	0.004626
Model:Technique	3	0.05545	0.018483	40.83196	1E-25
Measure:Model:Technique	6	0.004512	0.000752	1.661262	0.126556
Residuals	2376	1.075547	0.000453		

Table G.12: Three-way ANOVA for Dataset 12.

Hamming-loss

Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.012288	0.006144	13.76486	1.14E-06
Model	3	0.132978	0.044326	99.30532	1.48E-60
Technique	1	22.69425	22.69425	50842.94	0
Measure:Model	6	0.002396	0.000399	0.894816	0.497632
Measure:Technique	2	0.001322	0.000661	1.480893	0.227644
Model:Technique	3	0.044625	0.014875	33.32491	4.29E-21
Measure:Model:Technique	6	0.001026	0.000171	0.38298	0.890281
Residuals	2376	1.060551	0.000446		

One-error

Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.028972	0.014486	23.73867	6.2E-11
Model	3	0.359014	0.119671	196.111	1.2E-113
Technique	1	0.122999	0.122999	201.5635	5.68E-44
Measure:Model	6	0.008952	0.001492	2.445129	0.023267
Measure:Technique	2	0.020147	0.010073	16.5077	7.59E-08
Model:Technique	3	0.273223	0.091074	149.2475	1.31E-88
Measure:Model:Technique	6	0.002153	0.000359	0.588141	0.740081
Residuals	2376	1.449891	0.00061		

Precision

Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.005678	0.002839	1.724516	0.178482
Model	3	0.19083	0.06361	38.63665	2.25E-24
Technique	1	11.78214	11.78214	7156.454	0
Measure:Model	6	0.006175	0.001029	0.625103	0.710355
Measure:Technique	2	0.002936	0.001468	0.891571	0.410148
Model:Technique	3	0.019382	0.006461	3.924173	0.008306
Measure:Model:Technique	6	0.000671	0.000112	0.067916	0.998787
Residuals	2376	3.911765	0.001646		

Recall

Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.064706	0.032353	13.88055	1.02E-06
Model	3	0.402255	0.134085	57.52675	6.78E-36
Technique	1	3.429394	3.429394	1471.321	5.8E-251
Measure:Model	6	0.010899	0.001817	0.779343	0.586068
Measure:Technique	2	0.000151	7.53E-05	0.032308	0.968209
Model:Technique	3	0.295135	0.098378	42.20755	1.44E-26
Measure:Model:Technique	6	0.004728	0.000788	0.338077	0.916984
Residuals	2376	5.538044	0.002331		

Table G.13: Three-way ANOVA for Dataset 13.

Hamming-loss

Dataset 13	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000375	0.000187	10.53883	2.77E-05
Model	3	0.218795	0.072932	4104.215	0
Technique	1	11.81594	11.81594	664939.9	0
Measure:Model	6	0.002301	0.000383	21.578	8.23E-25
Measure:Technique	2	0.000186	9.3E-05	5.230867	0.005411
Model:Technique	3	0.011691	0.003897	219.2956	1.4E-125
Measure:Model:Technique	6	0.000211	3.51E-05	1.974807	0.065886
Residuals	2376	0.042221	1.78E-05		

One-error

Dataset 13	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001309	0.000654	14.68441	4.59E-07
Model	3	0.134655	0.044885	1007.246	0
Technique	1	0.10137	0.10137	2274.804	0
Measure:Model	6	0.000917	0.000153	3.429468	0.00225
Measure:Technique	2	0.001045	0.000523	11.73069	8.52E-06
Model:Technique	3	0.015395	0.005132	115.1549	1.27E-69
Measure:Model:Technique	6	0.000184	3.07E-05	0.688679	0.658824
Residuals	2376	0.10588	4.46E-05		

Precision

Dataset 13	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000835	0.000418	10.44865	3.03E-05
Model	3	0.191993	0.063998	1601.142	0
Technique	1	35.74047	35.74047	894184	0
Measure:Model	6	0.002596	0.000433	10.82574	6.47E-12
Measure:Technique	2	0.000907	0.000453	11.34377	1.25E-05
Model:Technique	3	0.019059	0.006353	158.9426	6.93E-94
Measure:Model:Technique	6	0.00032	5.33E-05	1.332546	0.238942
Residuals	2376	0.094969	4E-05		

Recall

Dataset 13	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001155	0.000577	7.603939	0.000511
Model	3	0.480452	0.160151	2108.996	0
Technique	1	0.012175	0.012175	160.3284	1.32E-35
Measure:Model	6	0.003554	0.000592	7.801361	2.47E-08
Measure:Technique	2	0.000434	0.000217	2.8582	0.057569
Model:Technique	3	0.000315	0.000105	1.380918	0.246736
Measure:Model:Technique	6	0.000504	8.39E-05	1.105232	0.35665
Residuals	2376	0.180426	7.59E-05		

Table G.14: Three-way ANOVA for Dataset 14.

Hamming-loss

Dataset 14	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.285771	0.142886	185.0223	2.09E-75
Model	3	2.348638	0.782879	1013.748	0
Technique	1	4.433552	4.433552	5740.994	0
Measure:Model	6	0.120358	0.02006	25.97518	4.58E-30
Measure:Technique	2	0.006418	0.003209	4.155164	0.015797
Model:Technique	3	1.73946	0.57982	750.8074	0
Measure:Model:Technique	6	0.007508	0.001251	1.620384	0.137379
Residuals	2376	1.834895	0.000772		

One-error

Dataset 14	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.412046	0.206023	56.28143	1.31E-24
Model	3	8.702583	2.900861	792.4587	0
Technique	1	6.04668	6.04668	1651.835	1.2E-274
Measure:Model	6	0.209532	0.034922	9.539989	2.19E-10
Measure:Technique	2	0.012908	0.006454	1.76312	0.171733
Model:Technique	3	5.484604	1.828201	499.429	1.3E-251
Measure:Model:Technique	6	0.010474	0.001746	0.476868	0.82597
Residuals	2376	8.697546	0.003661		

Precision

Dataset 14	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.41733	0.208665	114.9808	2.17E-48
Model	3	2.460248	0.820083	451.8904	2.8E-232
Technique	1	14.47325	14.47325	7975.201	0
Measure:Model	6	0.184546	0.030758	16.9484	2.96E-19
Measure:Technique	2	0.004323	0.002161	1.191002	0.304098
Model:Technique	3	1.747554	0.582518	320.985	6E-175
Measure:Model:Technique	6	0.000898	0.00015	0.082505	0.997894
Residuals	2376	4.311923	0.001815		

Recall

Dataset 14	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.421474	0.210737	81.97621	3.74E-35
Model	3	4.656816	1.552272	603.83	1.1E-291
Technique	1	8.59717	8.59717	3344.279	0
Measure:Model	6	0.175683	0.029281	11.39007	1.37E-12
Measure:Technique	2	0.029747	0.014874	5.785817	0.003114
Model:Technique	3	3.196169	1.06539	414.434	1.7E-216
Measure:Model:Technique	6	0.034626	0.005771	2.244919	0.036519
Residuals	2376	6.108007	0.002571		

Table G.15: Three-way ANOVA for Dataset 15.

Hamming-loss

Dataset 15	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000232	0.000116	2.352008	0.095399
Model	3	0.202356	0.067452	1369.833	0
Technique	1	47.4348	47.4348	963317.3	0
Measure:Model	6	0.000399	6.64E-05	1.348873	0.23185
Measure:Technique	2	3.98E-05	1.99E-05	0.404517	0.667345
Model:Technique	3	0.028413	0.009471	192.3418	1.2E-111
Measure:Model:Technique	6	0.000128	2.13E-05	0.432909	0.857317
Residuals	2376	0.116997	4.92E-05		

One-error

Dataset 15	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.00123	0.000615	4.273051	0.014046
Model	3	0.743351	0.247784	1721.237	0
Technique	1	0.049559	0.049559	344.2611	7.08E-72
Measure:Model	6	0.000583	9.72E-05	0.674928	0.669974
Measure:Technique	2	0.00071	0.000355	2.464429	0.085275
Model:Technique	3	0.387755	0.129252	897.8508	0
Measure:Model:Technique	6	0.000122	2.03E-05	0.141079	0.990753
Residuals	2376	0.342041	0.000144		

Precision

Dataset 15	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001171	0.000586	4.088121	0.016889
Model	3	0.230353	0.076784	536.0084	5.2E-266
Technique	1	80.59754	80.59754	562627.5	0
Measure:Model	6	0.000906	0.000151	1.053662	0.388432
Measure:Technique	2	0.000512	0.000256	1.788756	0.167393
Model:Technique	3	0.031827	0.010609	74.0591	8.65E-46
Measure:Model:Technique	6	0.000496	8.27E-05	0.577171	0.74883
Residuals	2376	0.340367	0.000143		

Recall

Dataset 15	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001558	0.000779	2.518005	0.080835
Model	3	0.863162	0.287721	930.0513	0
Technique	1	2.208973	2.208973	7140.461	0
Measure:Model	6	0.00172	0.000287	0.926546	0.47454
Measure:Technique	2	0.001197	0.000598	1.934365	0.144743
Model:Technique	3	0.030171	0.010057	32.5089	1.37E-20
Measure:Model:Technique	6	0.000664	0.000111	0.357549	0.90578
Residuals	2376	0.735039	0.000309		

Table G.16: Three-way ANOVA for Dataset 16.

Hamming-loss

Dataset 16	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001948	0.000974	2.728104	0.065548
Model	3	0.750779	0.25026	700.8896	0
Technique	1	37.14033	37.14033	104017.1	0
Measure:Model	6	0.002099	0.00035	0.979616	0.437294
Measure:Technique	2	0.004601	0.002301	6.443421	0.001619
Model:Technique	3	0.567218	0.189073	529.527	1.7E-263
Measure:Model:Technique	6	0.002344	0.000391	1.094298	0.363231
Residuals	2376	0.848374	0.000357		

One-error

Dataset 16	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.031922	0.015961	8.568064	0.000196
Model	3	3.553504	1.184501	635.8512	2.2E-303
Technique	1	1.781807	1.781807	956.4903	8.7E-177
Measure:Model	6	0.006946	0.001158	0.621427	0.713325
Measure:Technique	2	0.019053	0.009526	5.113801	0.00608
Model:Technique	3	2.767028	0.922343	495.1221	6.9E-250
Measure:Model:Technique	6	0.008638	0.00144	0.772853	0.591206
Residuals	2376	4.426153	0.001863		

Precision

Dataset 16	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010261	0.005131	2.78782	0.061757
Model	3	0.825371	0.275124	149.4931	9.6E-89
Technique	1	40.00982	40.00982	21740.01	0
Measure:Model	6	0.00335	0.000558	0.303337	0.935415
Measure:Technique	2	0.009001	0.0045	2.445351	0.086914
Model:Technique	3	0.524885	0.174962	95.0683	4.16E-58
Measure:Model:Technique	6	0.004725	0.000788	0.427925	0.860744
Residuals	2376	4.372737	0.00184		

Recall

Dataset 16	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004016	0.002008	0.818815	0.441078
Model	3	2.747128	0.915709	373.3674	1.2E-198
Technique	1	0.264044	0.264044	107.6604	1.07E-24
Measure:Model	6	0.006922	0.001154	0.47042	0.830685
Measure:Technique	2	0.010516	0.005258	2.143942	0.117419
Model:Technique	3	2.274677	0.758226	309.1557	1.9E-169
Measure:Model:Technique	6	0.008923	0.001487	0.606341	0.725485
Residuals	2376	5.827304	0.002453		

Table G.17: Three-way ANOVA for Dataset 17.

Hamming-loss

Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001349	0.000675	13.50368	1.47E-06
Model	3	0.441512	0.147171	2946.25	0
Technique	1	8.882804	8.882804	177827.5	0
Measure:Model	6	0.004448	0.000741	14.84128	1E-16
Measure:Technique	2	9.09E-05	4.55E-05	0.909997	0.402666
Model:Technique	3	0.011902	0.003967	79.42551	5.8E-49
Measure:Model:Technique	6	6.57E-05	1.09E-05	0.219107	0.970803
Residuals	2376	0.118686	5E-05		

One-error

Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.008989	0.004495	21.29279	6.83E-10
Model	3	1.584177	0.528059	2501.619	0
Technique	1	0.94677	0.94677	4485.214	0
Measure:Model	6	0.016098	0.002683	12.71045	3.61E-14
Measure:Technique	2	0.000226	0.000113	0.534437	0.586069
Model:Technique	3	0.101302	0.033767	159.9692	1.93E-94
Measure:Model:Technique	6	0.000606	0.000101	0.478104	0.825062
Residuals	2376	0.501542	0.000211		

Precision

Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006909	0.003454	16.89201	5.19E-08
Model	3	0.38962	0.129873	635.0873	4.1E-303
Technique	1	43.77186	43.77186	214046.7	0
Measure:Model	6	0.011043	0.00184	8.99976	9.57E-10
Measure:Technique	2	0.001227	0.000613	2.999187	0.050016
Model:Technique	3	0.04629	0.01543	75.45368	1.29E-46
Measure:Model:Technique	6	0.001652	0.000275	1.346006	0.233083
Residuals	2376	0.485884	0.000204		

Recall

Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006592	0.003296	12.26811	5E-06
Model	3	2.579666	0.859889	3200.367	0
Technique	1	35.31979	35.31979	131454.5	0
Measure:Model	6	0.013276	0.002213	8.235395	7.64E-09
Measure:Technique	2	0.00437	0.002185	8.133044	0.000302
Model:Technique	3	0.096485	0.032162	119.7003	3.4E-72
Measure:Model:Technique	6	0.000736	0.000123	0.456657	0.840621
Residuals	2376	0.638394	0.000269		

Table G.18: Three-way ANOVA for Dataset 18.

Hamming-loss

Dataset 18	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.319207	0.159603	311.8379	5.5E-121
Model	3	0.169994	0.056665	110.713	4.25E-67
Technique	1	4.114322	4.114322	8038.693	0
Measure:Model	6	0.053499	0.008916	17.42131	8E-20
Measure:Technique	2	0.733488	0.366744	716.5562	3.1E-244
Model:Technique	3	0.060302	0.020101	39.27323	9.13E-25
Measure:Model:Technique	6	0.008083	0.001347	2.631979	0.015144
Residuals	2376	1.216072	0.000512		

One-error

Dataset 18	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	2.215519	1.10776	479.4818	1.2E-175
Model	3	1.124286	0.374762	162.2117	1.18E-95
Technique	1	0.359822	0.359822	155.745	1.15E-34
Measure:Model	6	0.27505	0.045842	19.84205	9.92E-23
Measure:Technique	2	0.502769	0.251384	108.8091	6.1E-46
Model:Technique	3	0.583421	0.194474	84.17575	9.34E-52
Measure:Model:Technique	6	0.071569	0.011928	5.162971	2.76E-05
Residuals	2376	5.489337	0.00231		

Precision

Dataset 18	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.525049	0.262524	162.7662	5.66E-67
Model	3	0.194804	0.064935	40.25972	2.26E-25
Technique	1	8.399185	8.399185	5207.531	0
Measure:Model	6	0.079181	0.013197	8.182137	8.83E-09
Measure:Technique	2	0.14064	0.07032	43.59856	2.54E-19
Model:Technique	3	0.051329	0.01711	10.60808	6.29E-07
Measure:Model:Technique	6	0.008912	0.001485	0.920898	0.478607
Residuals	2376	3.832231	0.001613		

Recall

Dataset 18	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	1.493291	0.746646	264.9831	1.3E-104
Model	3	0.72243	0.24081	85.463	1.64E-52
Technique	1	45.22039	45.22039	16048.63	0
Measure:Model	6	0.169231	0.028205	10.00993	6.06E-11
Measure:Technique	2	0.26326	0.13163	46.71517	1.26E-20
Model:Technique	3	0.159926	0.053309	18.91914	4E-12
Measure:Model:Technique	6	0.029771	0.004962	1.76096	0.103278
Residuals	2376	6.69488	0.002818		

Table G.19: Three-way ANOVA for Dataset 19.

Hamming-loss

Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004374	0.002187	59.58088	5.65E-26
Model	3	0.239709	0.079903	2176.65	0
Technique	1	24.83103	24.83103	676425.5	0
Measure:Model	6	0.004967	0.000828	22.55204	5.61E-26
Measure:Technique	2	0.000237	0.000119	3.230231	0.039722
Model:Technique	3	0.016972	0.005657	154.1111	2.9E-91
Measure:Model:Technique	6	0.000185	3.08E-05	0.839857	0.538943
Residuals	2376	0.087221	3.67E-05		

One-error

Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.020444	0.010222	40.92607	3.35E-18
Model	3	1.781592	0.593864	2377.689	0
Technique	1	1.98436	1.98436	7944.897	0
Measure:Model	6	0.015851	0.002642	10.5772	1.28E-11
Measure:Technique	2	0.001686	0.000843	3.375044	0.034381
Model:Technique	3	0.072444	0.024148	96.68231	4.84E-59
Measure:Model:Technique	6	0.000558	9.3E-05	0.372172	0.896983
Residuals	2376	0.593442	0.00025		

Precision

Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.020003	0.010002	37.75498	7.22E-17
Model	3	0.308954	0.102985	388.7516	2E-205
Technique	1	44.21112	44.21112	166890.5	0
Measure:Model	6	0.016039	0.002673	10.09113	4.85E-11
Measure:Technique	2	0.005676	0.002838	10.71277	2.34E-05
Model:Technique	3	0.017041	0.00568	21.44296	1.06E-13
Measure:Model:Technique	6	0.004629	0.000772	2.912381	0.007842
Residuals	2376	0.629428	0.000265		

Recall

Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.011456	0.005728	13.49368	1.49E-06
Model	3	2.051532	0.683844	1611.017	0
Technique	1	31.3984	31.3984	73969.13	0
Measure:Model	6	0.017727	0.002955	6.960394	2.38E-07
Measure:Technique	2	0.003412	0.001706	4.018995	0.018093
Model:Technique	3	0.008938	0.002979	7.018721	0.000107
Measure:Model:Technique	6	0.000599	9.98E-05	0.235086	0.965168
Residuals	2376	1.008564	0.000424		

Table G.20: Three-way ANOVA for Dataset 20.

Hamming-loss

Dataset 20	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.02514	0.01257	34.83818	1.22E-15
Model	3	0.035965	0.011988	33.226	4.94E-21
Technique	1	21.97061	21.97061	60892.04	0
Measure:Model	6	0.006156	0.001026	2.843632	0.009228
Measure:Technique	2	0.000171	8.56E-05	0.237189	0.788861
Model:Technique	3	0.007301	0.002434	6.745177	0.000158
Measure:Model:Technique	6	0.00165	0.000275	0.762368	0.599534
Residuals	2376	0.85729	0.000361		

One-error

Dataset 20	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.045539	0.022769	25.82901	7.99E-12
Model	3	0.151561	0.05052	57.30899	9.17E-36
Technique	1	0.646272	0.646272	733.1122	6E-141
Measure:Model	6	0.028929	0.004821	5.469302	1.24E-05
Measure:Technique	2	0.027032	0.013516	15.33193	2.42E-07
Model:Technique	3	0.074232	0.024744	28.06875	7.83E-18
Measure:Model:Technique	6	0.014495	0.002416	2.740467	0.011761
Residuals	2376	2.094552	0.000882		

Precision

Dataset 20	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.083611	0.041805	20.54271	1.43E-09
Model	3	0.059211	0.019737	9.698635	2.32E-06
Technique	1	16.45629	16.45629	8086.462	0
Measure:Model	6	0.012519	0.002086	1.025273	0.406724
Measure:Technique	2	0.023561	0.01178	5.788737	0.003105
Model:Technique	3	0.002023	0.000674	0.33132	0.802713
Measure:Model:Technique	6	0.002812	0.000469	0.230327	0.966901
Residuals	2376	4.83526	0.002035		

Recall

Dataset 20	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.038762	0.019381	6.337256	0.001799
Model	3	0.214718	0.071573	23.40298	6.31E-15
Technique	1	31.58112	31.58112	10326.46	0
Measure:Model	6	0.037321	0.00622	2.033864	0.058026
Measure:Technique	2	0.047825	0.023913	7.818965	0.000412
Model:Technique	3	0.053433	0.017811	5.823902	0.000582
Measure:Model:Technique	6	0.010033	0.001672	0.546795	0.772806
Residuals	2376	7.266452	0.003058		

Table G.21: Three-way ANOVA for Dataset 21.

Hamming-loss

Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	5.03E-05	2.51E-05	1.664946	0.189422
Model	3	0.174062	0.058021	3843.013	0
Technique	1	24.08499	24.08499	1595277	0
Measure:Model	6	3.5E-05	5.83E-06	0.385982	0.88839
Measure:Technique	2	6.76E-05	3.38E-05	2.238065	0.10689
Model:Technique	3	0.039417	0.013139	870.2747	0
Measure:Model:Technique	6	7.5E-05	1.25E-05	0.828473	0.54769
Residuals	2376	0.035872	1.51E-05		

One-error

Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000222	0.000111	1.884006	0.152207
Model	3	0.1625	0.054167	920.2066	0
Technique	1	0.243704	0.243704	4140.159	0
Measure:Model	6	6.38E-05	1.06E-05	0.18057	0.982223
Measure:Technique	2	0.000134	6.69E-05	1.136558	0.321096
Model:Technique	3	0.023538	0.007846	133.2909	8.26E-80
Measure:Model:Technique	6	0.000128	2.13E-05	0.361682	0.903326
Residuals	2376	0.13986	5.89E-05		

Precision

Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000127	6.36E-05	1.829564	0.16071
Model	3	0.134617	0.044872	1290.923	0
Technique	1	97.63838	97.63838	2808940	0
Measure:Model	6	0.000102	1.71E-05	0.491337	0.815261
Measure:Technique	2	8.71E-05	4.35E-05	1.252608	0.285947
Model:Technique	3	0.061943	0.020648	594.0112	4.7E-288
Measure:Model:Technique	6	2.67E-05	4.46E-06	0.128222	0.992858
Residuals	2376	0.082589	3.48E-05		

Recall

Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000189	9.47E-05	0.864808	0.421265
Model	3	0.630595	0.210198	1920.056	0
Technique	1	1.130271	1.130271	10324.45	0
Measure:Model	6	2.61E-05	4.36E-06	0.039799	0.99974
Measure:Technique	2	0.000666	0.000333	3.03971	0.048035
Model:Technique	3	0.027625	0.009208	84.113	1.02E-51
Measure:Model:Technique	6	0.000531	8.85E-05	0.808468	0.563199
Residuals	2376	0.260113	0.000109		

Table G.22: Three-way ANOVA for Dataset 22.

Hamming-loss

Dataset 22	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.081167	0.040583	82.59191	2.1E-35
Model	3	0.933131	0.311044	633.0107	2.3E-302
Technique	1	15.75735	15.75735	32068.07	0
Measure:Model	6	0.018099	0.003016	6.138857	2.12E-06
Measure:Technique	2	0.001808	0.000904	1.839565	0.159113
Model:Technique	3	0.799649	0.26655	542.4602	1.6E-268
Measure:Model:Technique	6	0.002434	0.000406	0.825663	0.549858
Residuals	2376	1.1675	0.000491		

One-error

Dataset 22	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.228944	0.114472	31.93929	2.05E-14
Model	3	8.377933	2.792644	779.1882	0
Technique	1	3.506502	3.506502	978.3646	3.6E-180
Measure:Model	6	0.059618	0.009936	2.772393	0.010913
Measure:Technique	2	0.044633	0.022317	6.226649	0.002008
Model:Technique	3	6.636218	2.212073	617.2003	1.3E-296
Measure:Model:Technique	6	0.013287	0.002214	0.617872	0.716195
Residuals	2376	8.515688	0.003584		

Precision

Dataset 22	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.156015	0.078007	37.52132	9.05E-17
Model	3	0.952618	0.317539	152.7355	1.63E-90
Technique	1	61.12853	61.12853	29402.66	0
Measure:Model	6	0.034189	0.005698	2.740826	0.011751
Measure:Technique	2	0.017021	0.00851	4.093453	0.016799
Model:Technique	3	0.931989	0.310663	149.428	1.04E-88
Measure:Model:Technique	6	0.005407	0.000901	0.433435	0.856954
Residuals	2376	4.939736	0.002079		

Recall

Dataset 22	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.197605	0.098802	26.03191	6.55E-12
Model	3	3.10428	1.03476	272.6331	4.6E-152
Technique	1	14.60046	14.60046	3846.852	0
Measure:Model	6	0.062273	0.010379	2.734538	0.011925
Measure:Technique	2	0.05676	0.02838	7.477433	0.000579
Model:Technique	3	2.327059	0.775686	204.3737	6.4E-118
Measure:Model:Technique	6	0.006547	0.001091	0.287483	0.943118
Residuals	2376	9.017945	0.003795		

Table G.23: Three-way ANOVA for Dataset 23.

Hamming-loss

Dataset 23	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.123792	0.061896	452.0451	4.3E-167
Model	3	0.15029	0.050097	365.8701	2.5E-195
Technique	1	45.55898	45.55898	332731	0
Measure:Model	6	0.155227	0.025871	188.9449	4E-197
Measure:Technique	2	0.027154	0.013577	99.15807	4.36E-42
Model:Technique	3	0.051768	0.017256	126.0247	9.43E-76
Measure:Model:Technique	6	0.03055	0.005092	37.18603	2.62E-43
Residuals	2376	0.325332	0.000137		

One-error

Dataset 23	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.297797	0.148898	333.7572	1.8E-128
Model	3	0.392275	0.130758	293.0962	7.2E-162
Technique	1	0.062073	0.062073	139.1364	3.01E-31
Measure:Model	6	0.388028	0.064671	144.9616	5.9E-157
Measure:Technique	2	0.002256	0.001128	2.528741	0.079974
Model:Technique	3	0.208702	0.069567	155.9359	2.96E-92
Measure:Model:Technique	6	0.002092	0.000349	0.781643	0.58425
Residuals	2376	1.06	0.000446		

Precision

Dataset 23	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.447254	0.223627	376.868	7E-143
Model	3	0.353088	0.117696	198.3478	8.5E-115
Technique	1	121.2788	121.2788	204385.5	0
Measure:Model	6	0.52669	0.087782	147.9344	9E-160
Measure:Technique	2	0.23015	0.115075	193.9303	9.65E-79
Model:Technique	3	0.241026	0.080342	135.3964	5.59E-81
Measure:Model:Technique	6	0.256134	0.042689	71.94171	1.26E-82
Residuals	2376	1.409877	0.000593		

Recall

Dataset 23	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.56386	0.28193	298.9632	1.5E-116
Model	3	0.70137	0.23379	247.915	5.9E-140
Technique	1	0.179908	0.179908	190.7778	8.48E-42
Measure:Model	6	0.730151	0.121692	129.0441	1.3E-141
Measure:Technique	2	0.187866	0.093933	99.60807	2.88E-42
Model:Technique	3	0.311084	0.103695	109.9595	1.14E-66
Measure:Model:Technique	6	0.21963	0.036605	38.81652	3.26E-45
Residuals	2376	2.240627	0.000943		

Table G.24: Three-way ANOVA for Dataset 24.

Hamming-loss

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.029252	0.014626	46.36516	1.77E-20
Model	3	0.401451	0.133817	424.2098	1.1E-220
Technique	1	37.34267	37.34267	118379	0
Measure:Model	6	0.003992	0.000665	2.109171	0.049265
Measure:Technique	2	0.010154	0.005077	16.0944	1.14E-07
Model:Technique	3	0.329104	0.109701	347.7617	3.3E-187
Measure:Model:Technique	6	0.001712	0.000285	0.904292	0.490674
Residuals	2376	0.749509	0.000315		

One-error

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.103064	0.051532	28.47363	6.02E-13
Model	3	3.011235	1.003745	554.611	3.5E-273
Technique	1	1.257398	1.257398	694.7648	1.6E-134
Measure:Model	6	0.047106	0.007851	4.337977	0.000232
Measure:Technique	2	0.044895	0.022448	12.40321	4.38E-06
Model:Technique	3	2.628356	0.876119	484.0921	1.9E-245
Measure:Model:Technique	6	0.016414	0.002736	1.511592	0.170237
Residuals	2376	4.300128	0.00181		

Precision

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.037191	0.018596	10.04129	4.54E-05
Model	3	0.431163	0.143721	77.60674	6.87E-48
Technique	1	73.14939	73.14939	39499.39	0
Measure:Model	6	0.005587	0.000931	0.502825	0.806638
Measure:Technique	2	0.040493	0.020247	10.93285	1.88E-05
Model:Technique	3	0.316111	0.10537	56.89815	1.62E-35
Measure:Model:Technique	6	0.003215	0.000536	0.289376	0.942223
Residuals	2376	4.400143	0.001852		

Recall

Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.206496	0.103248	31.78795	2.38E-14
Model	3	2.068368	0.689456	212.2695	5.5E-122
Technique	1	5.313725	5.313725	1635.988	1.3E-272
Measure:Model	6	0.018757	0.003126	0.962494	0.449115
Measure:Technique	2	0.006462	0.003231	0.994807	0.369949
Model:Technique	3	1.774922	0.591641	182.1541	2.6E-106
Measure:Model:Technique	6	0.007812	0.001302	0.400853	0.878844
Residuals	2376	7.717301	0.003248		

APPENDIX H

H.1 Comparing performance of techniques at different signal-to-noise ratios

Table H.1: Structure of Dataset 1 and Dataset 3.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 3	10	10	6	0	100	0.4	80	10 000

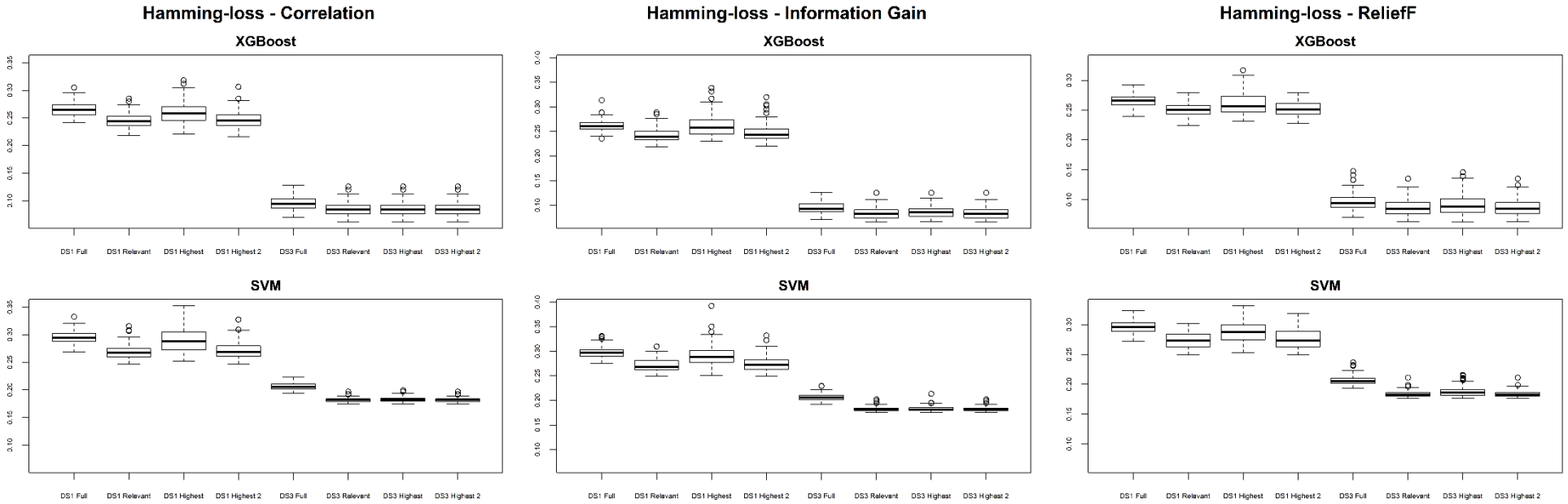


Figure H.1 Hamming-loss: Dataset 1 vs Dataset 3.

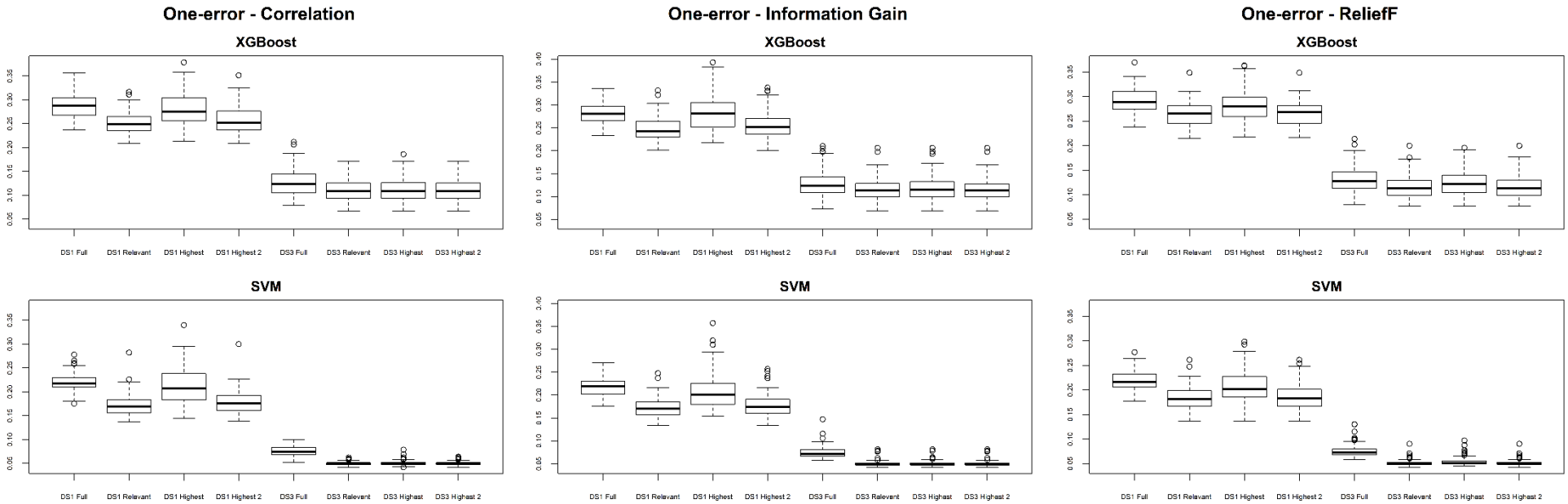


Figure H.2 One-error: Dataset 1 vs Dataset 3.

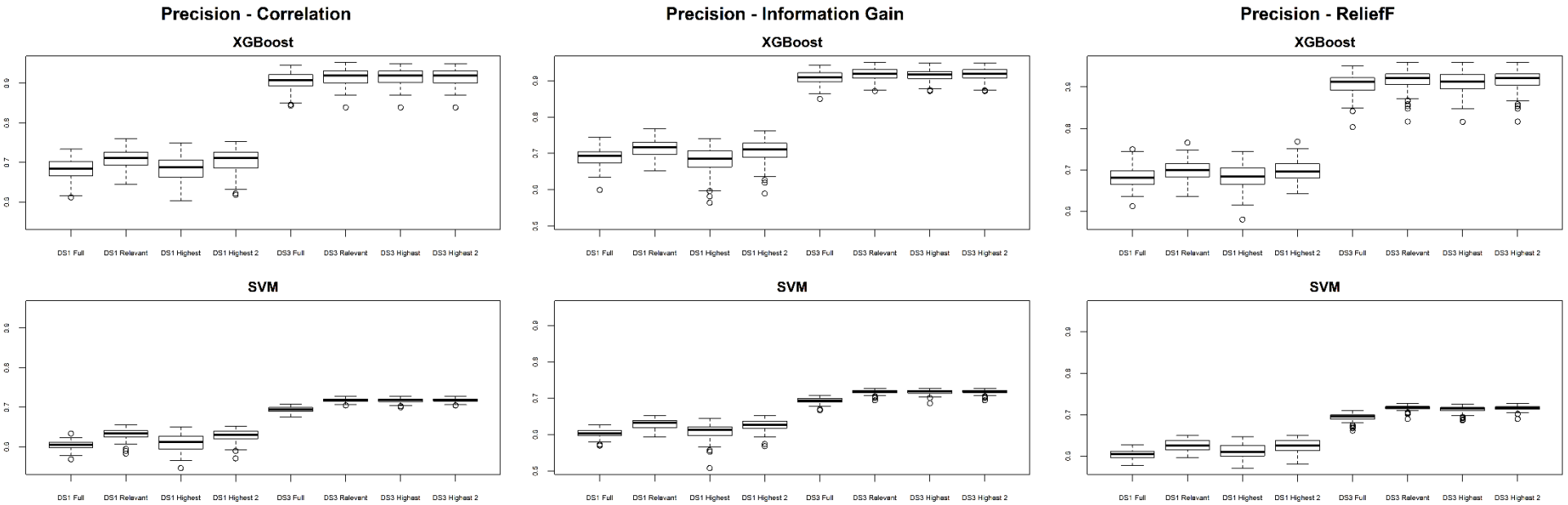


Figure H.3 Precision: Dataset 1 vs Dataset 3.

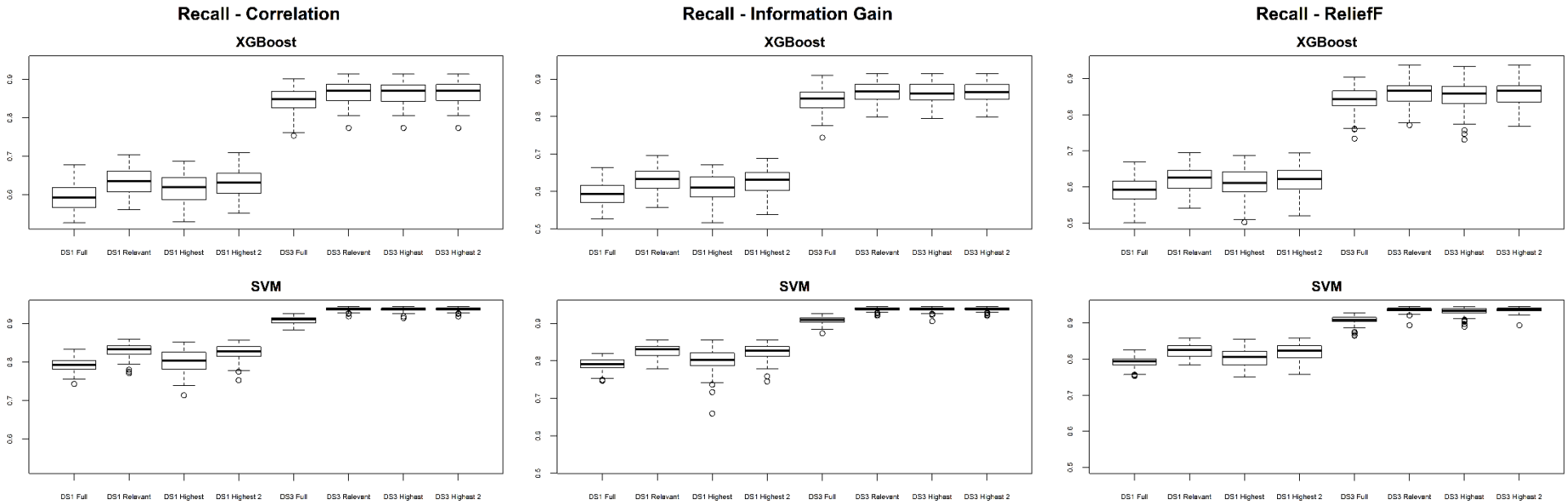


Figure H.4 Recall: Dataset 1 vs Dataset 3.

Table H.2: Structure of Dataset 17 and Dataset 21.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 17	50	10	6	0	10	vary	240	10 000
Dataset 21	50	10	6	0	100	vary	240	10 000

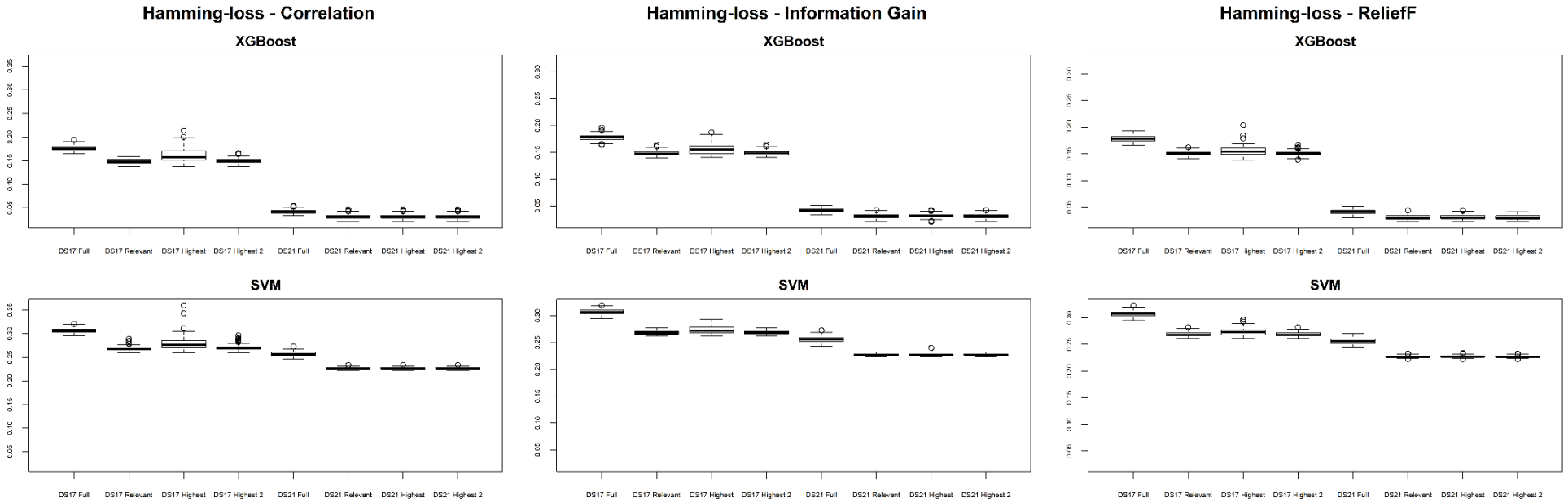


Figure H.5 Hamming-loss: Dataset 17 vs Dataset 21.

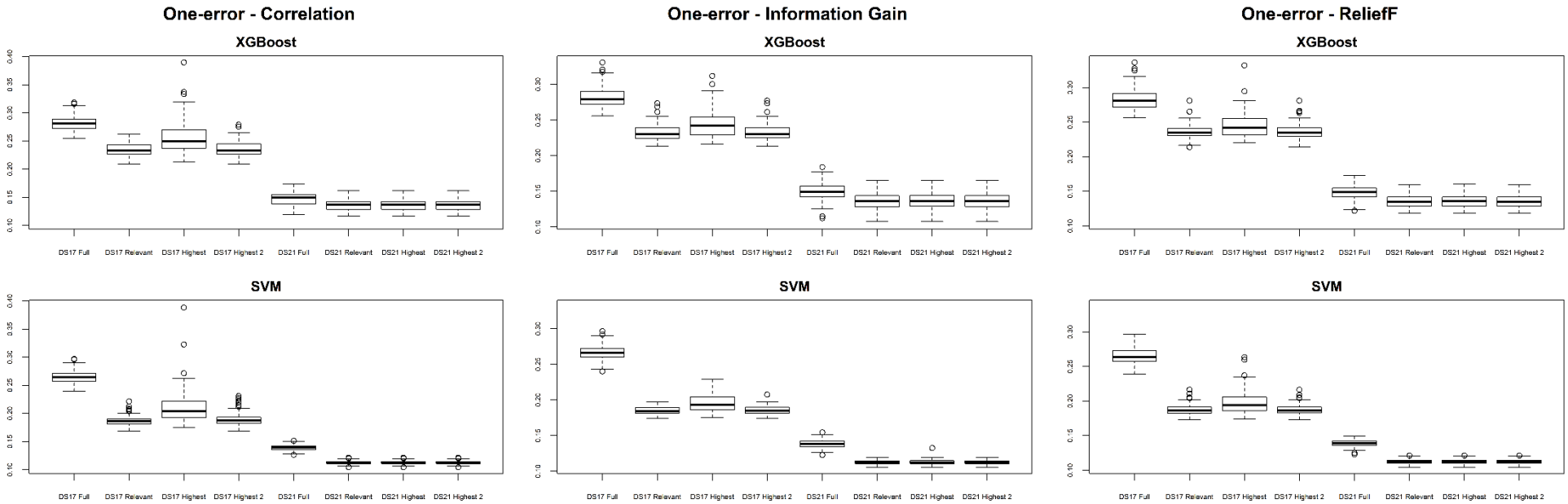


Figure H.6 One-error: Dataset 17 vs Dataset 21.

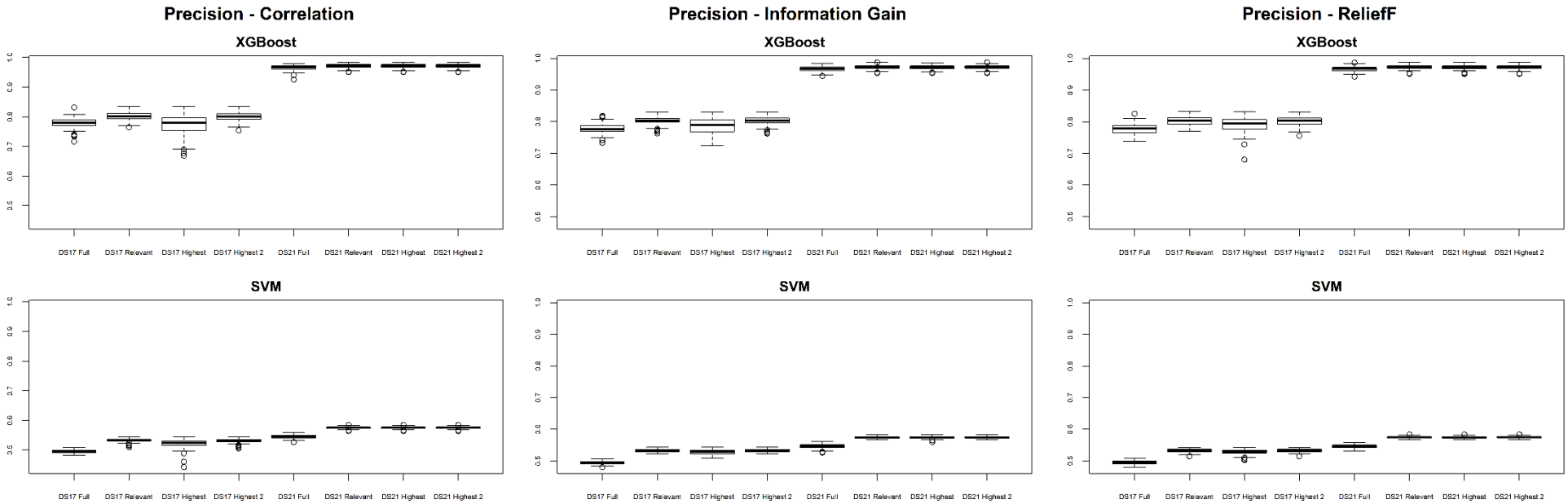


Figure H.7 Precision: Dataset 17 vs Dataset 21.

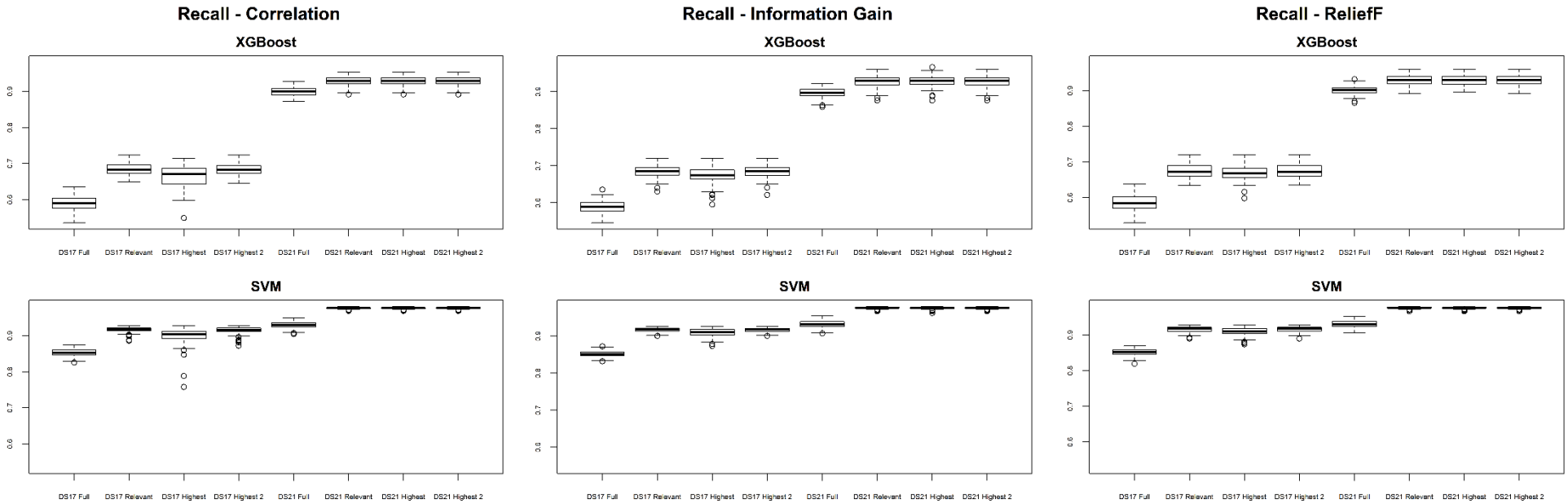


Figure H.8 Recall: Dataset 17 vs Dataset 21.

H.2 Comparing performance of techniques with respect to number of irrelevant features

Table H.3: Structure of Dataset 1 and Dataset 9.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 9	50	10	6	0	10	0.4	240	10 000

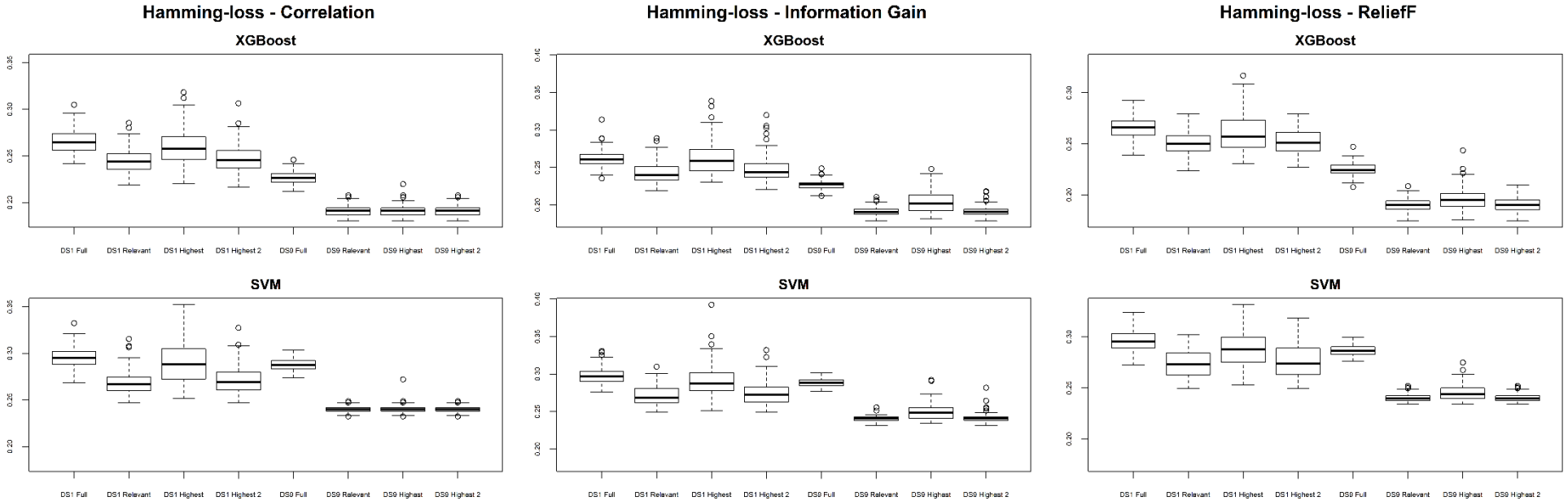


Figure H.9 Hamming-loss: Dataset 1 vs Dataset 9.

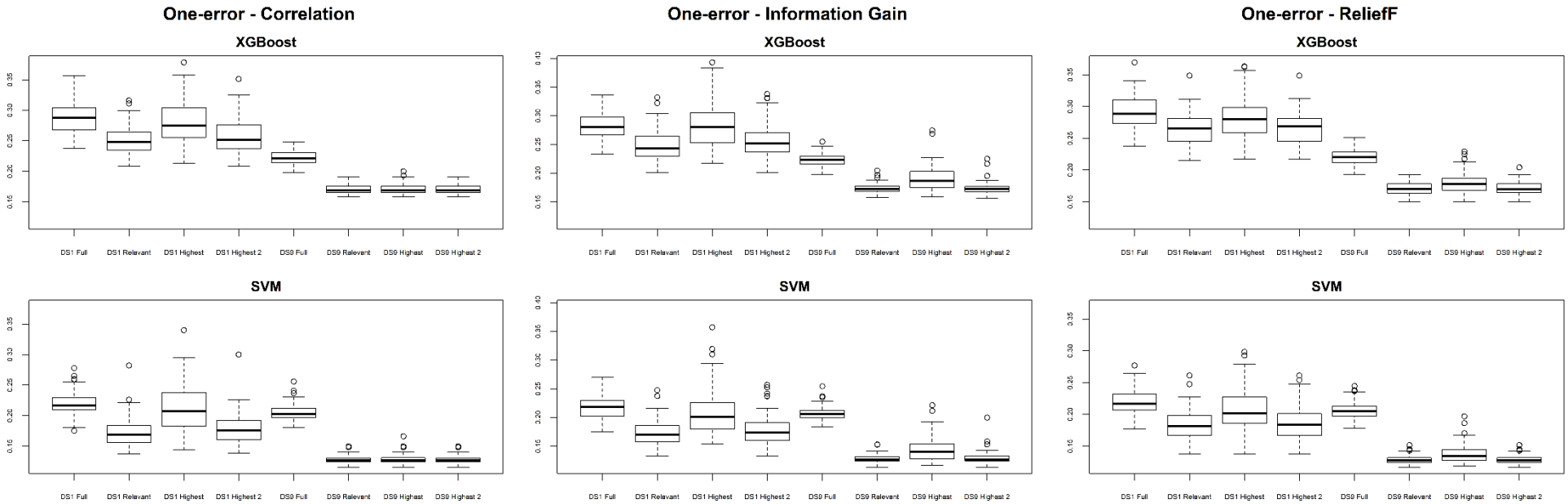


Figure H.10 One-error: Dataset 1 vs Dataset 9.

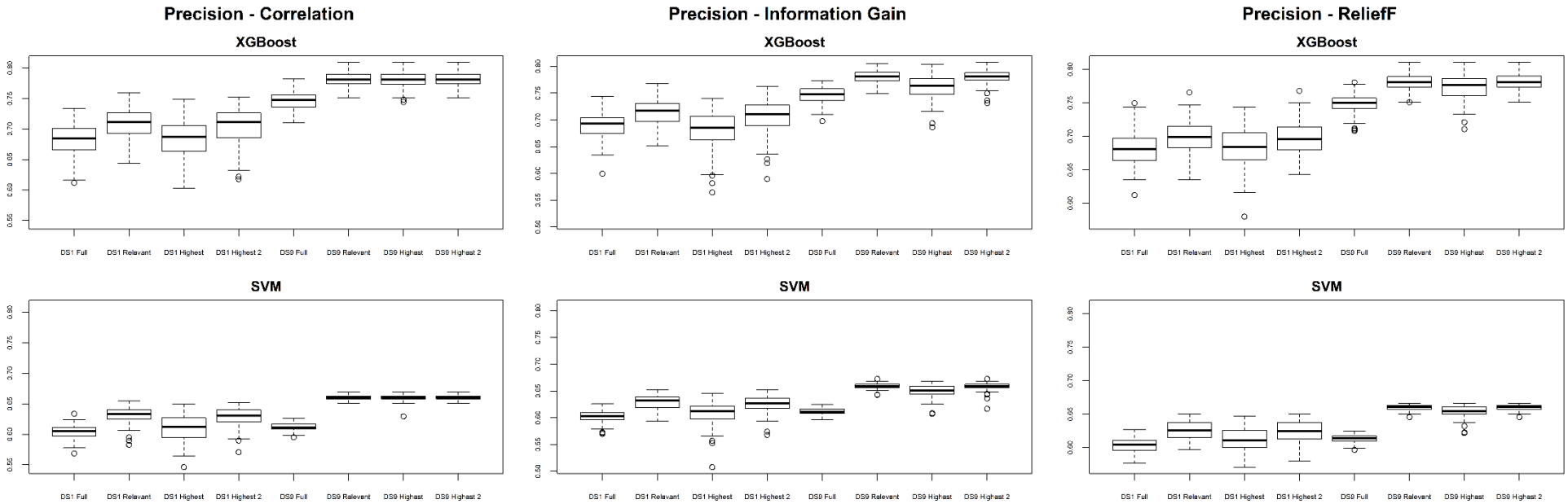


Figure H.11 Precision: Dataset 1 vs Dataset 9.

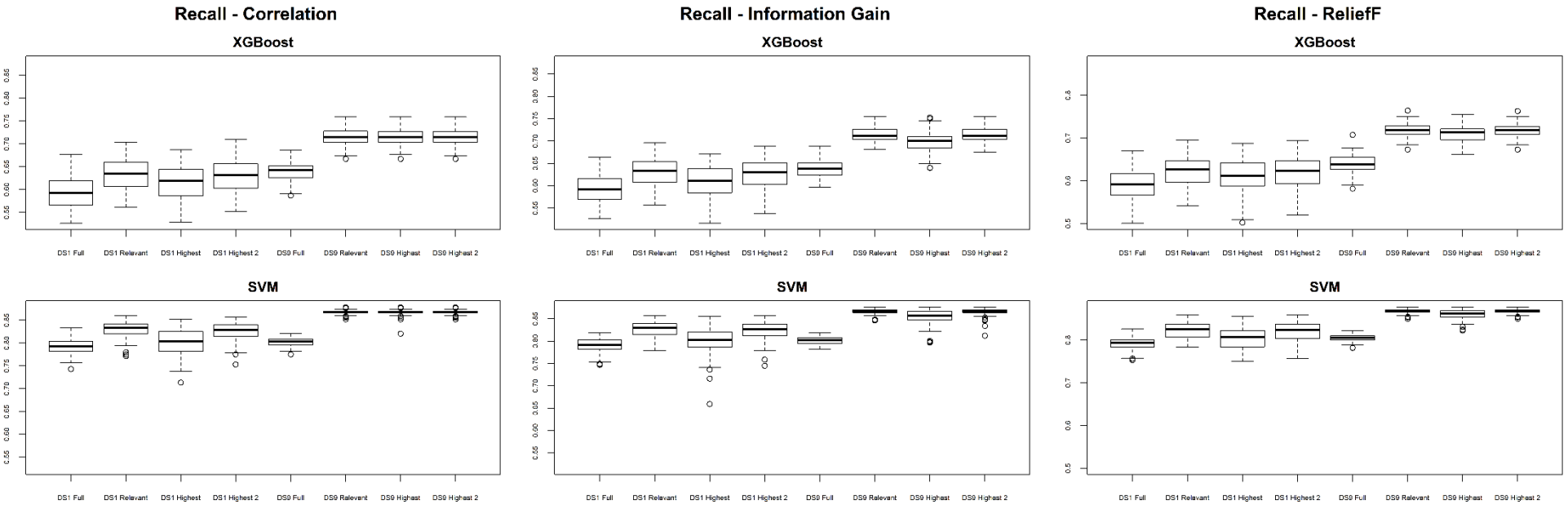


Figure H.12 Recall: Dataset 1 vs Dataset 9.

Table H.4: Structure of Dataset 5 and Dataset 17.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 5	10	10	6	0	10	vary	80	10 000
Dataset 17	50	10	6	0	10	vary	240	10 000

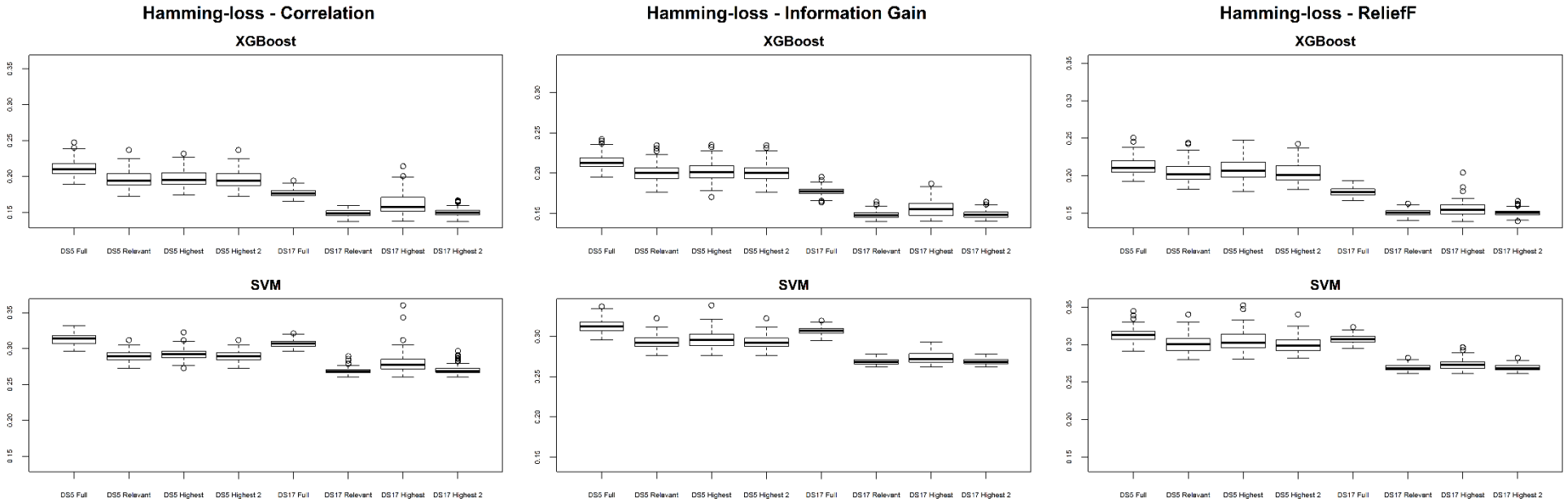


Figure H.13 Hamming-loss: Dataset 5 vs Dataset 17.

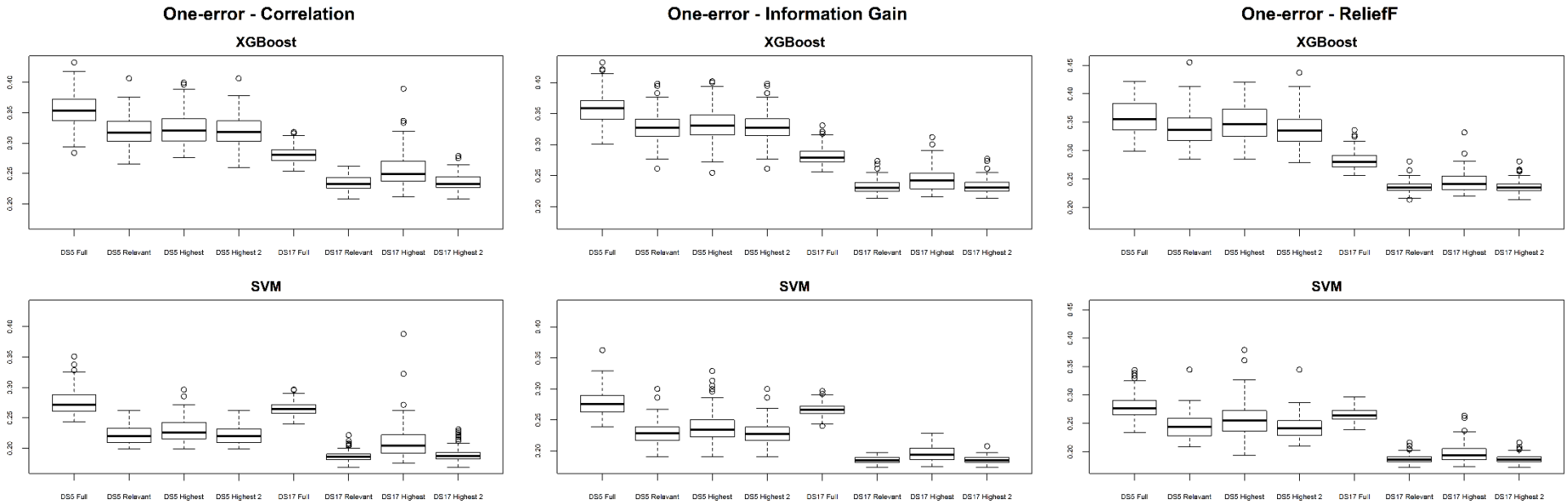


Figure H.14 One-error: Dataset 5 vs Dataset 17.

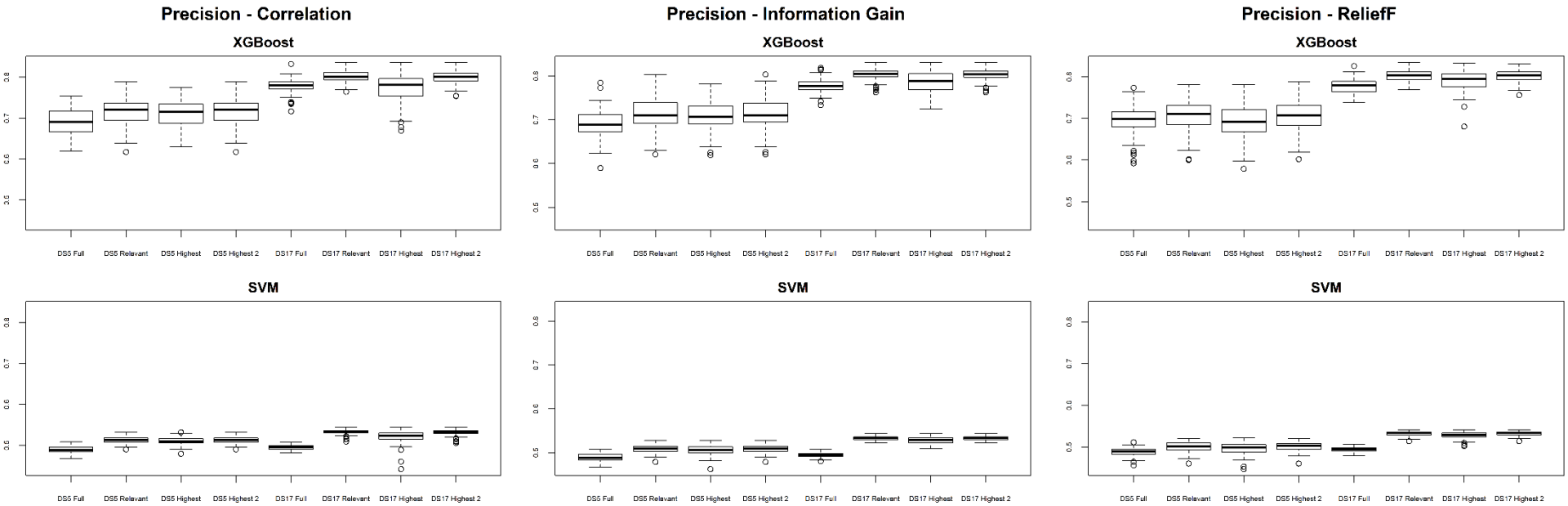


Figure H.15 Precision: Dataset 5 vs Dataset 17.

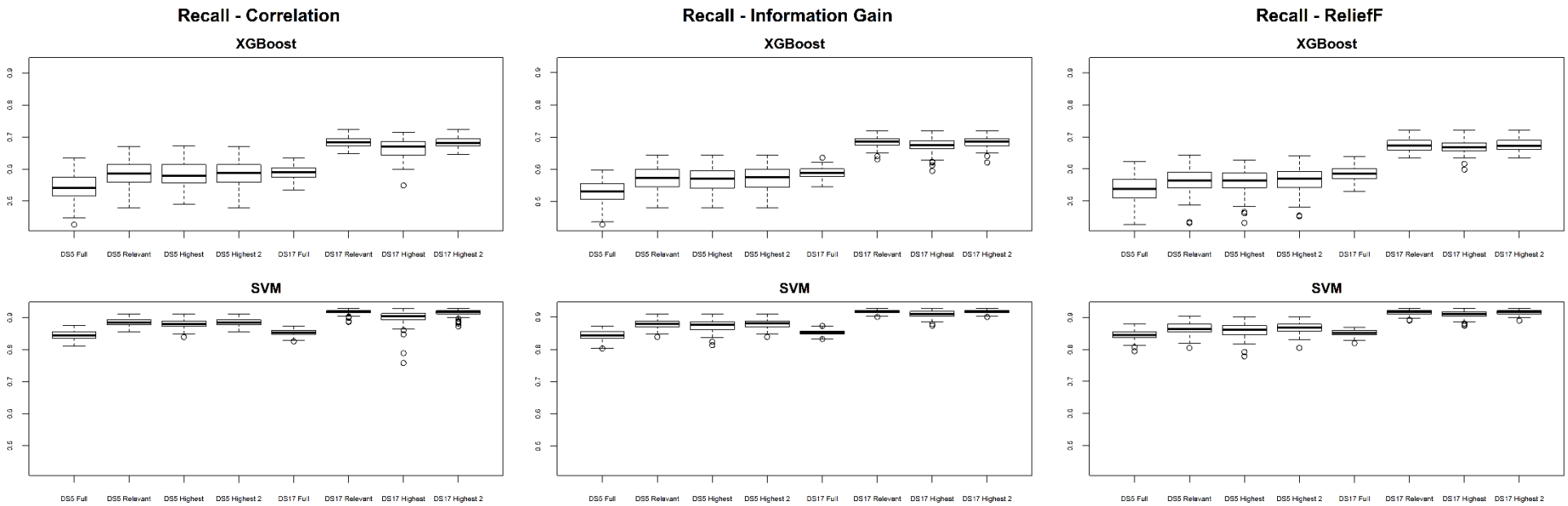


Figure H.16 Recall: Dataset 5 vs Dataset 17.

H.3 Comparing performance of techniques with respect to the label dependence (correlation)

Table H.5: Structure of Dataset 1 and Dataset 2.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 1	10	10	6	0	10	0.4	80	10 000
Dataset 2	10	10	6	0.4	10	0.4	80	10 000

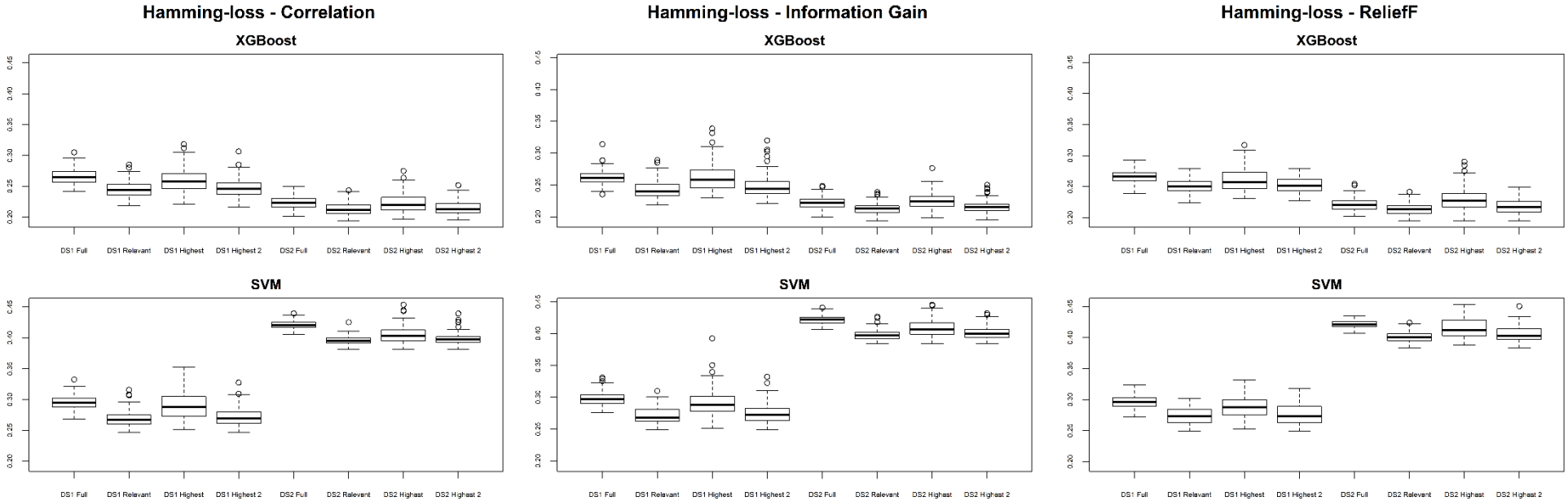


Figure H.17 Hamming-loss: Dataset 1 vs Dataset 2.

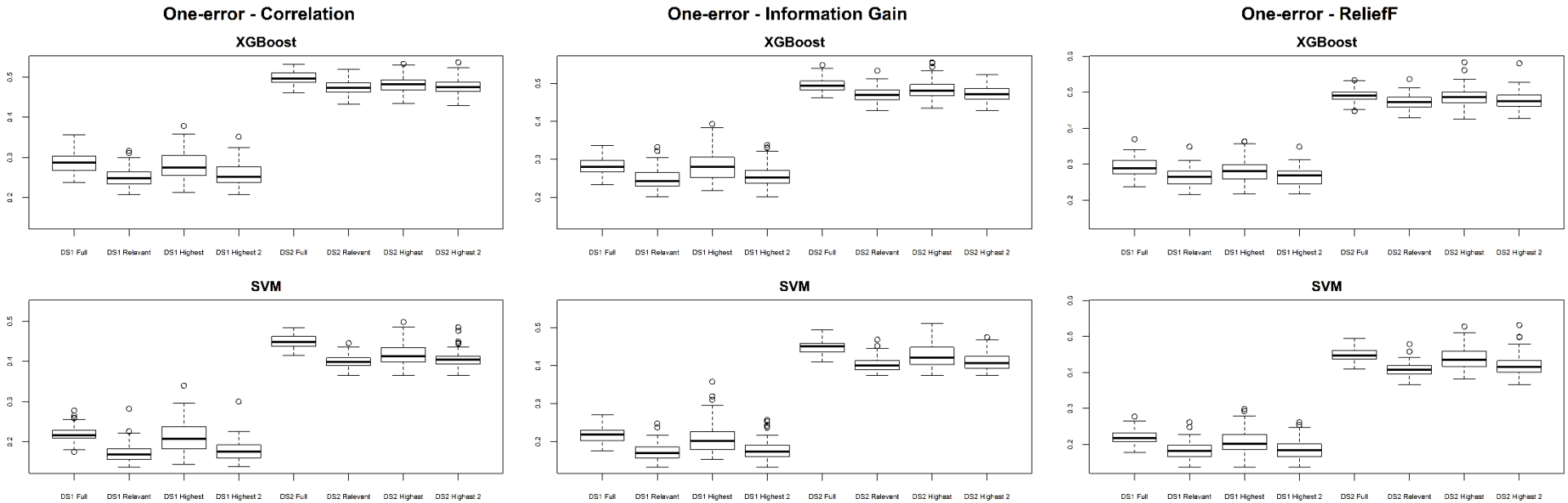


Figure H.18 One-error: Dataset 1 vs Dataset 2.

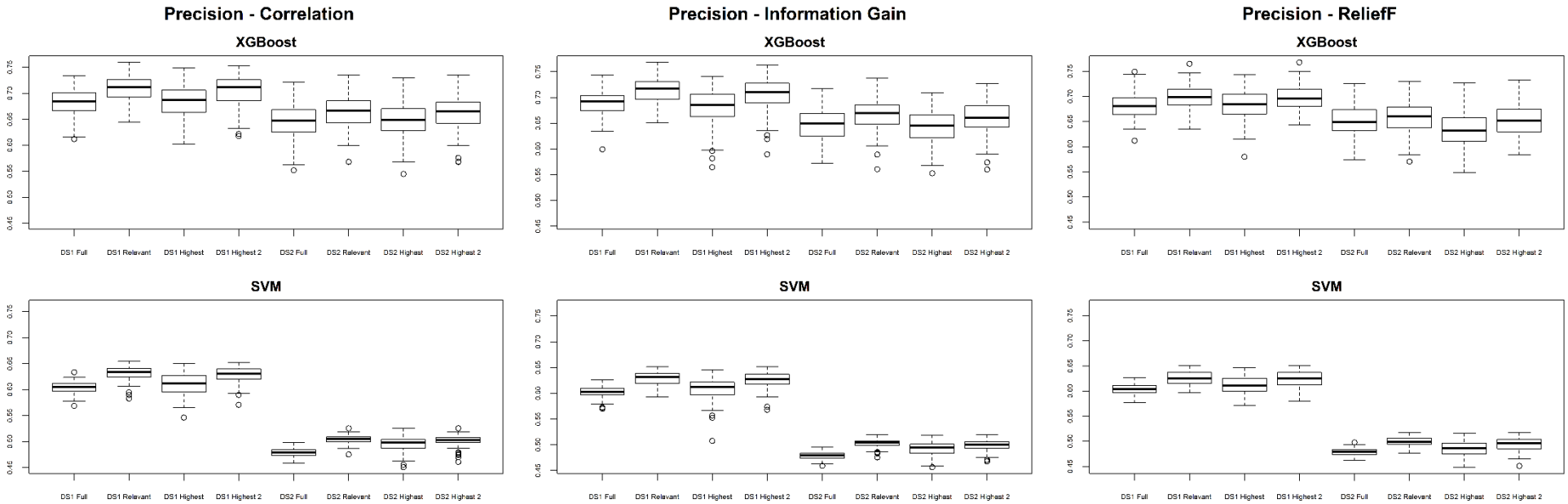


Figure H.19 Precision: Dataset 1 vs Dataset 2.

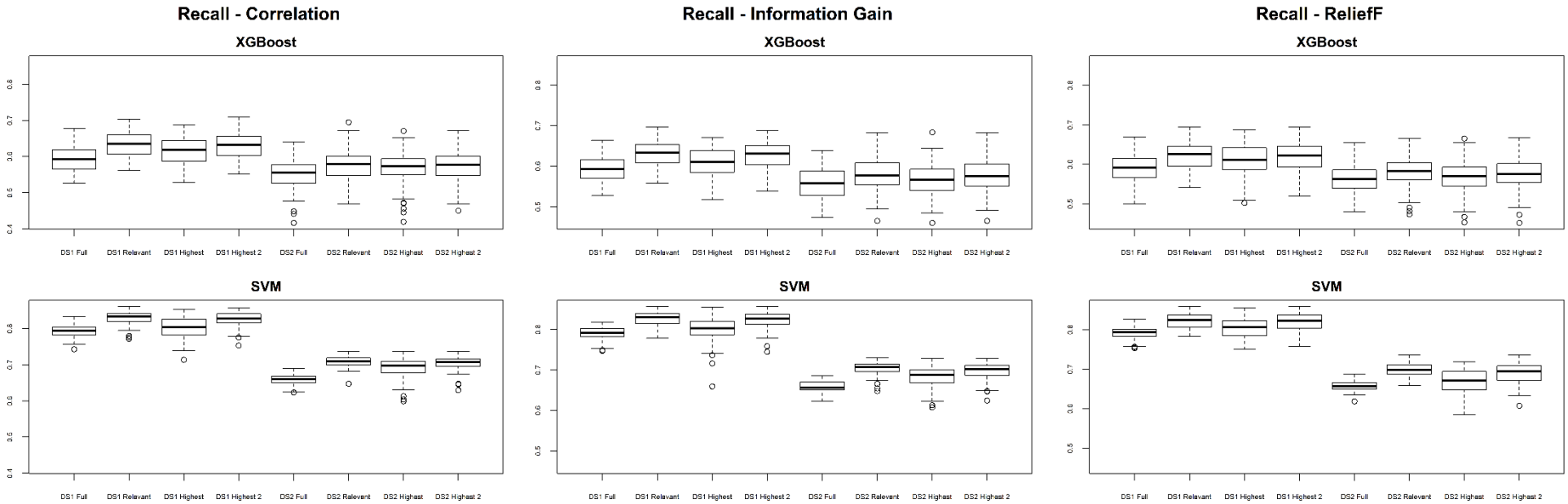


Figure H.20 Recall: Dataset 1 vs Dataset 2.

Table H.6: Structure of Dataset 10 and Dataset 12.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 10	50	10	6	0	10	0.4	30	10 000
Dataset 12	50	10	6	0.4	10	0.4	30	10 000

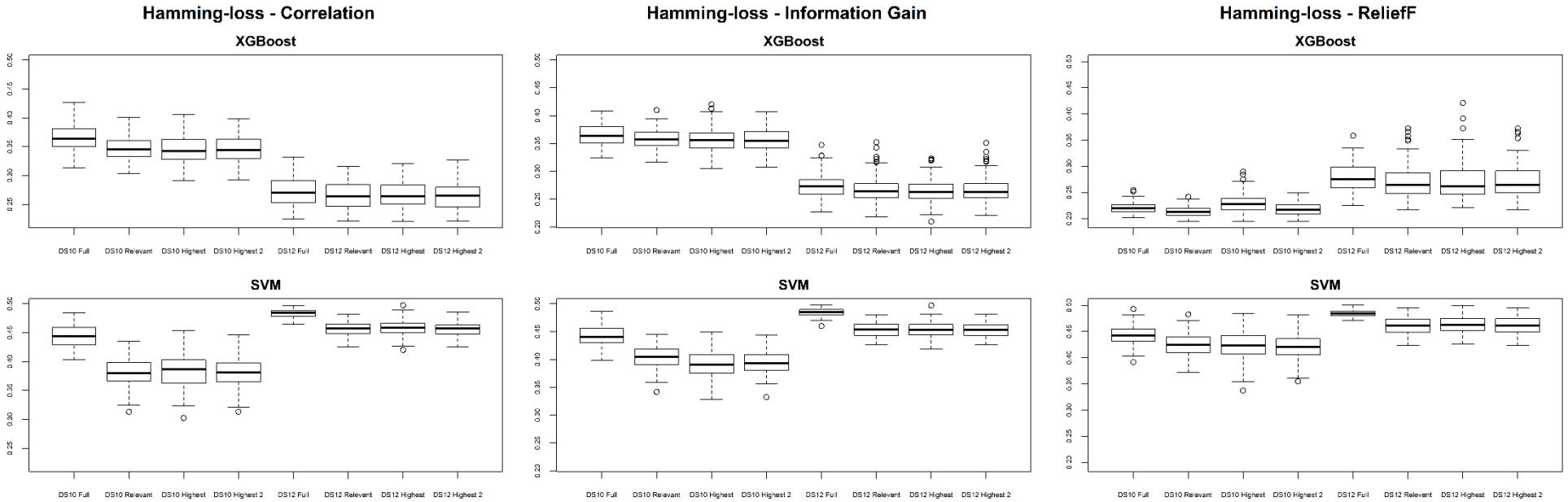


Figure H.21 Hamming-loss: Dataset 10 vs Dataset 12.

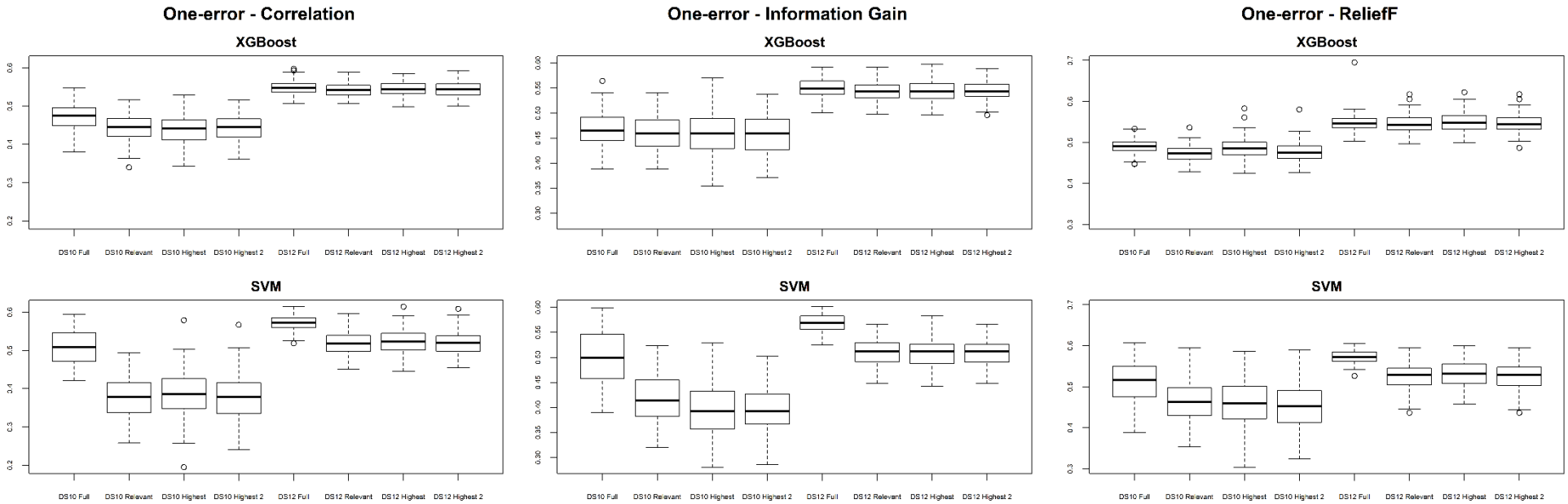


Figure H.22 One-error: Dataset 10 vs Dataset 12.

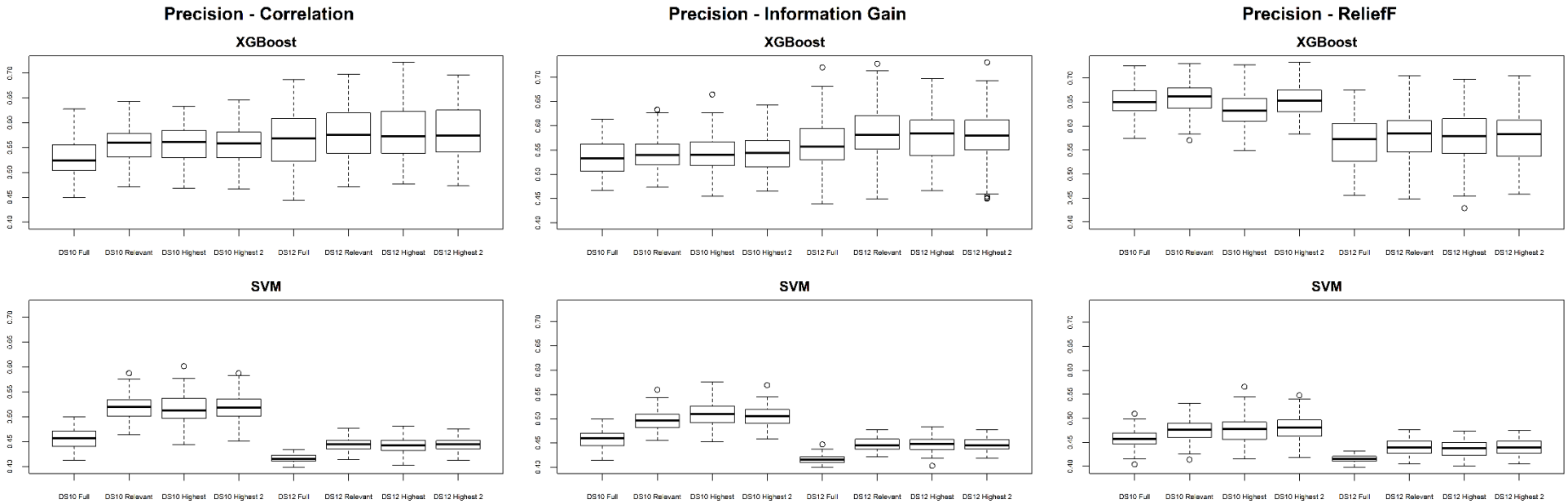


Figure H.23 Precision: Dataset 10 vs Dataset 12.

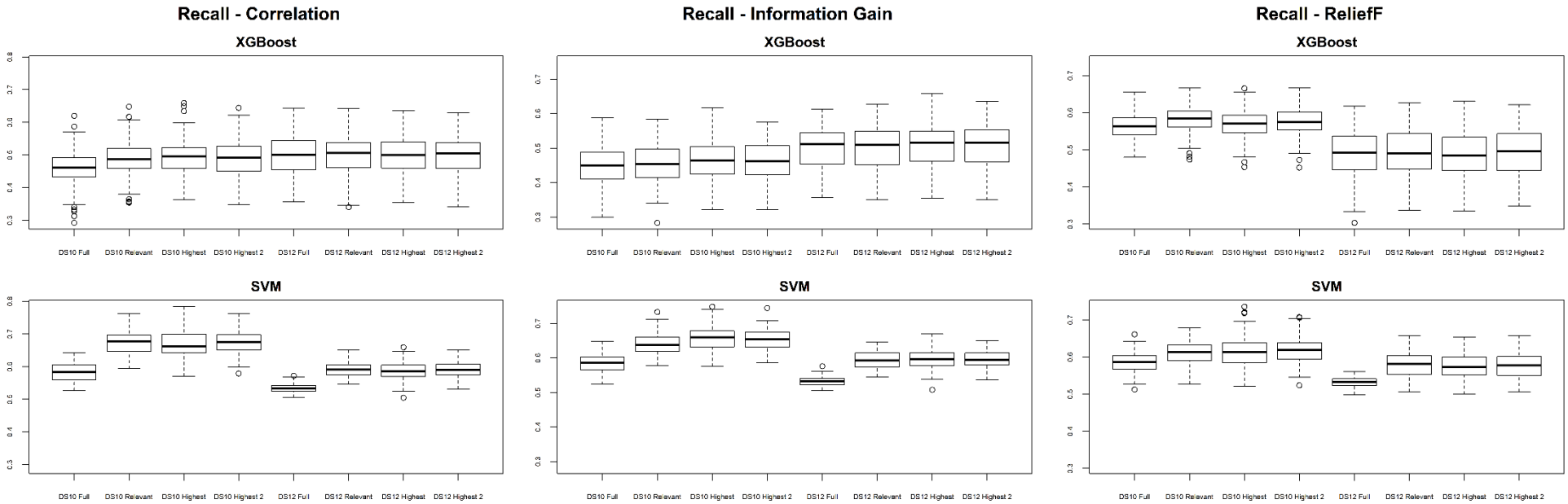


Figure H.24 Recall: Dataset 10 vs Dataset 12.

Table H.7: Structure of Dataset 17 and Dataset 19.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 17	50	10	6	0	10	vary	240	10 000
Dataset 19	50	10	6	0.4	10	vary	240	10 000

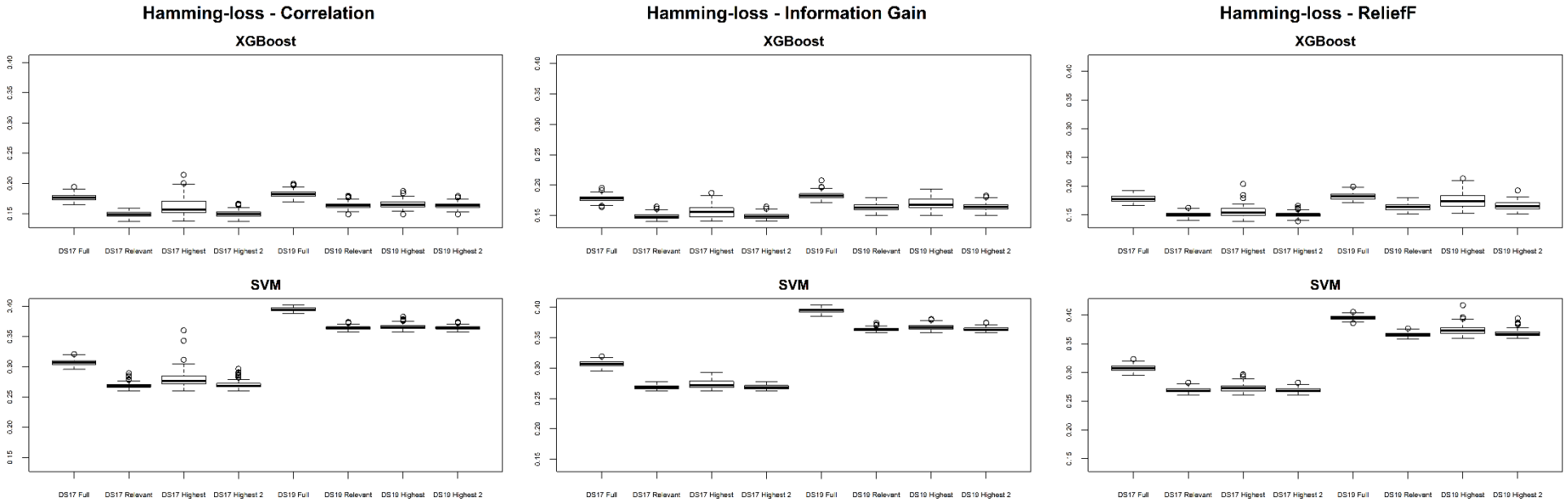


Figure H.25 Hamming-loss: Dataset 17 vs Dataset 19.

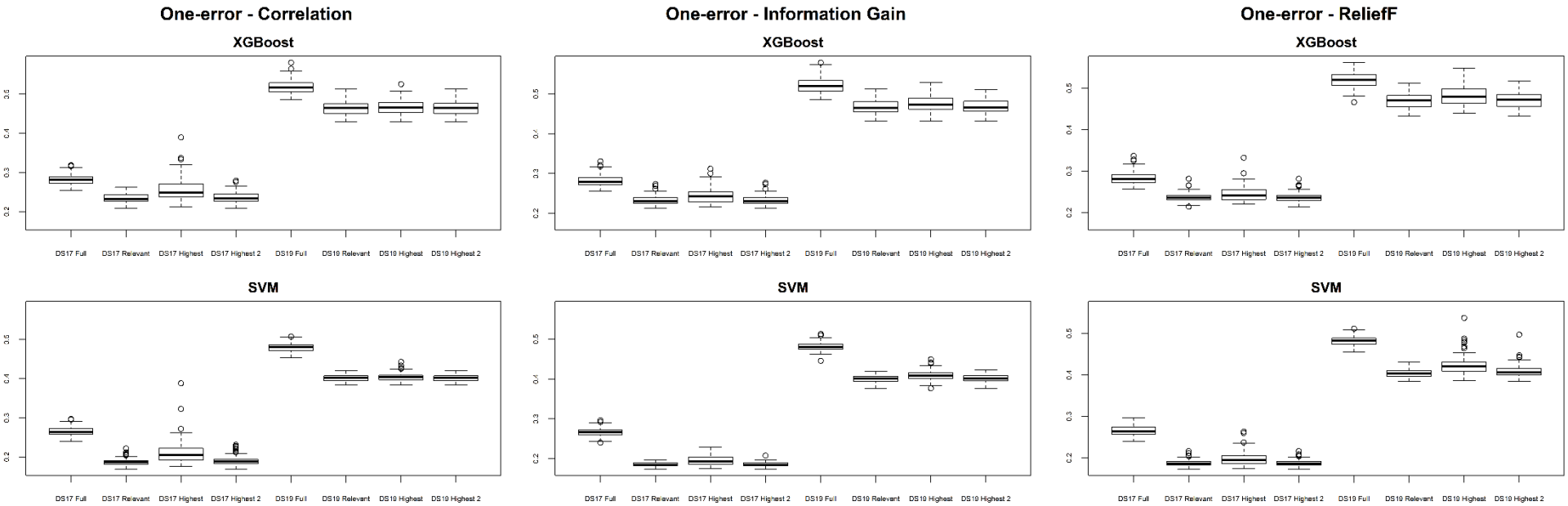


Figure H.26 One-error: Dataset 17 vs Dataset 19.

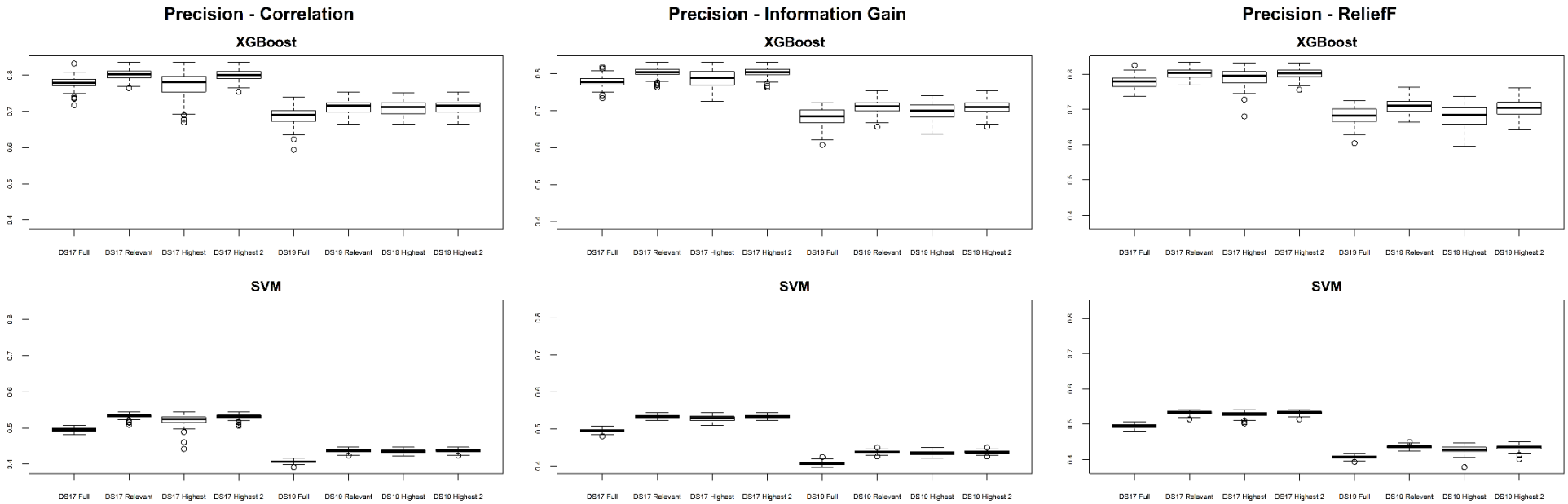


Figure H.27 Precision: Dataset 17 vs Dataset 19.

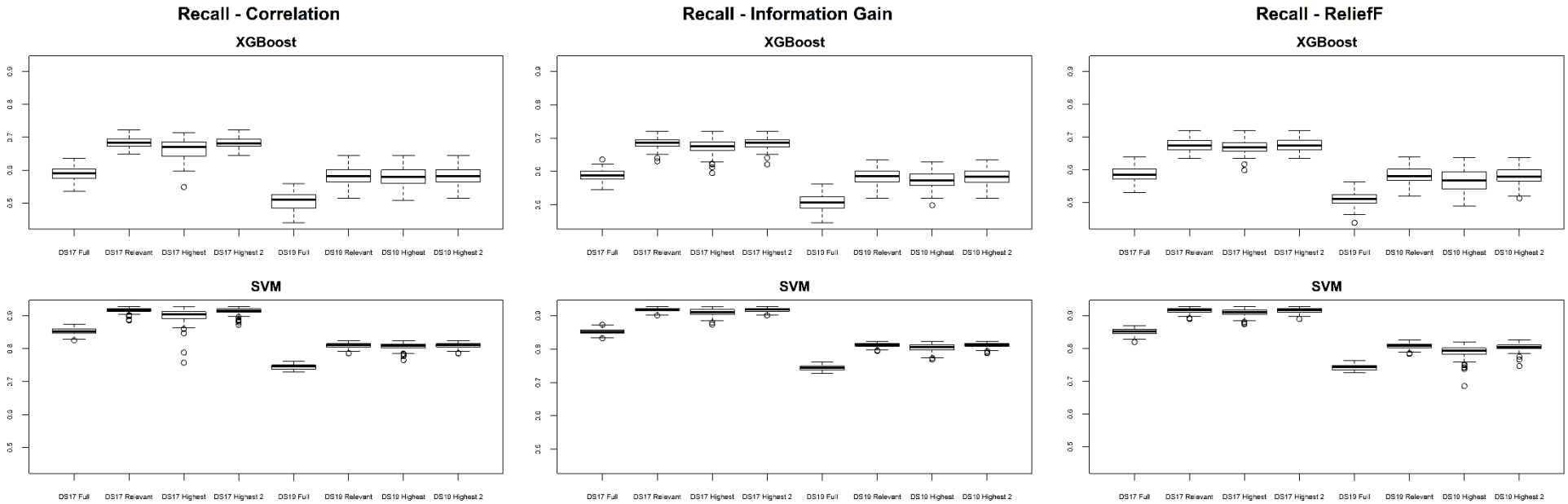


Figure H.28 Recall: Dataset 17 vs Dataset 19.

H.4 Comparing performance of techniques with respect to different density vectors

Table H.8: Structure of Dataset 9 and Dataset 17.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 9	50	10	6	0	10	0.4	240	10 000
Dataset 17	50	10	6	0	10	vary	240	10 000

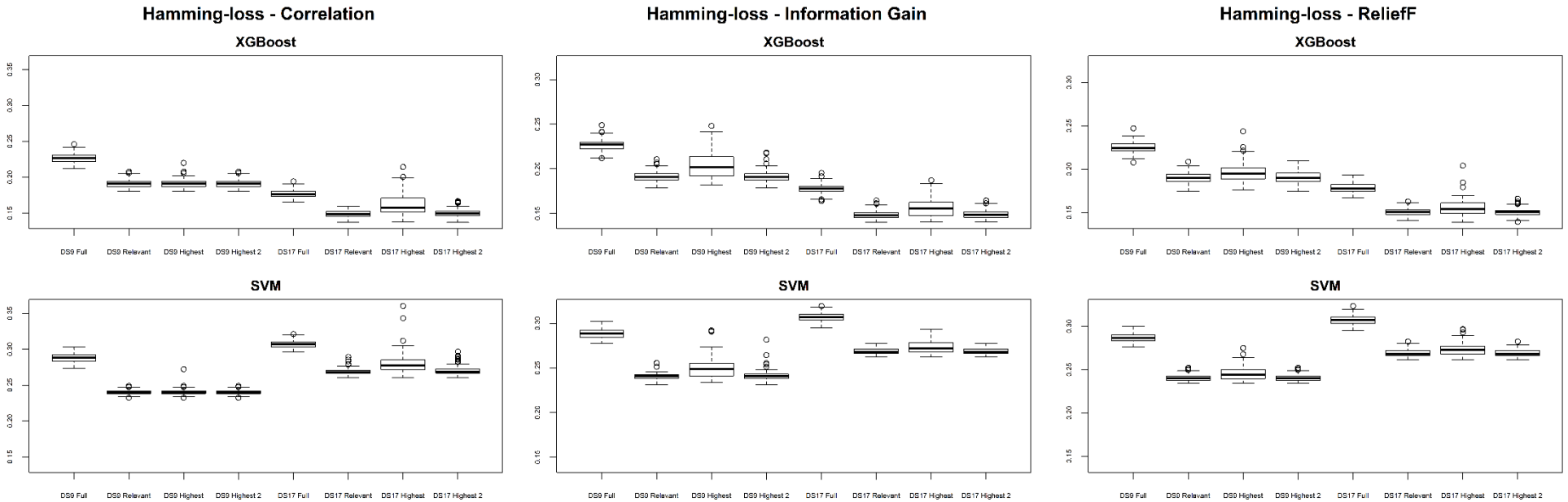


Figure H.29 Hamming-loss: Dataset 9 vs Dataset 17.

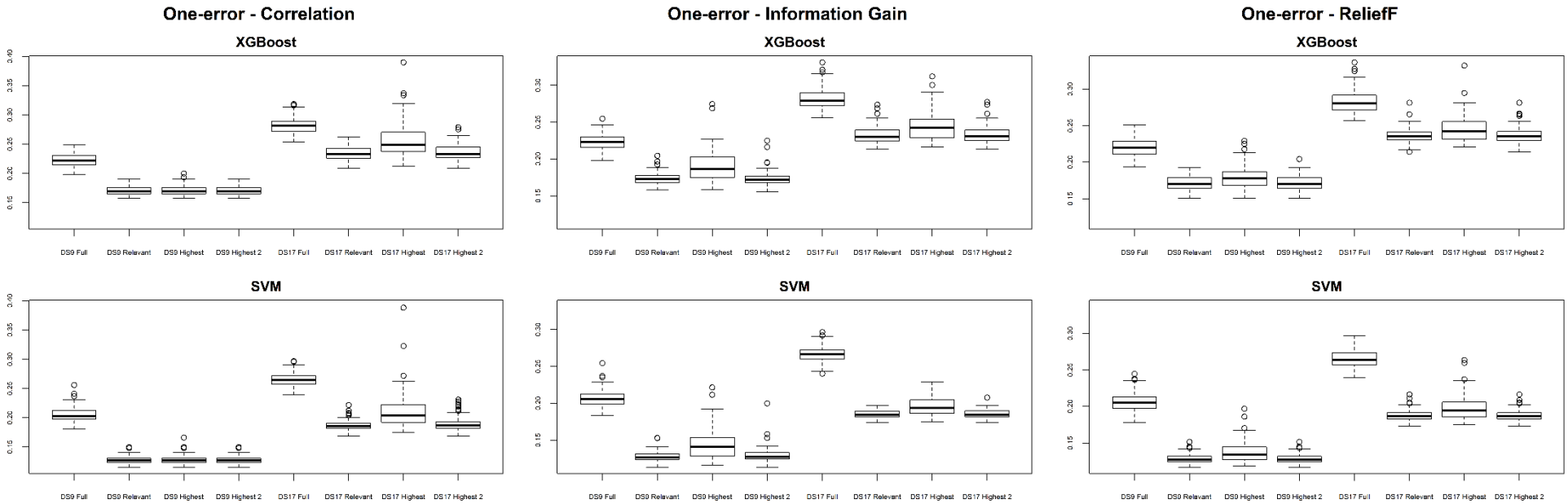


Figure H.30 One-error: Dataset 9 vs Dataset 17.

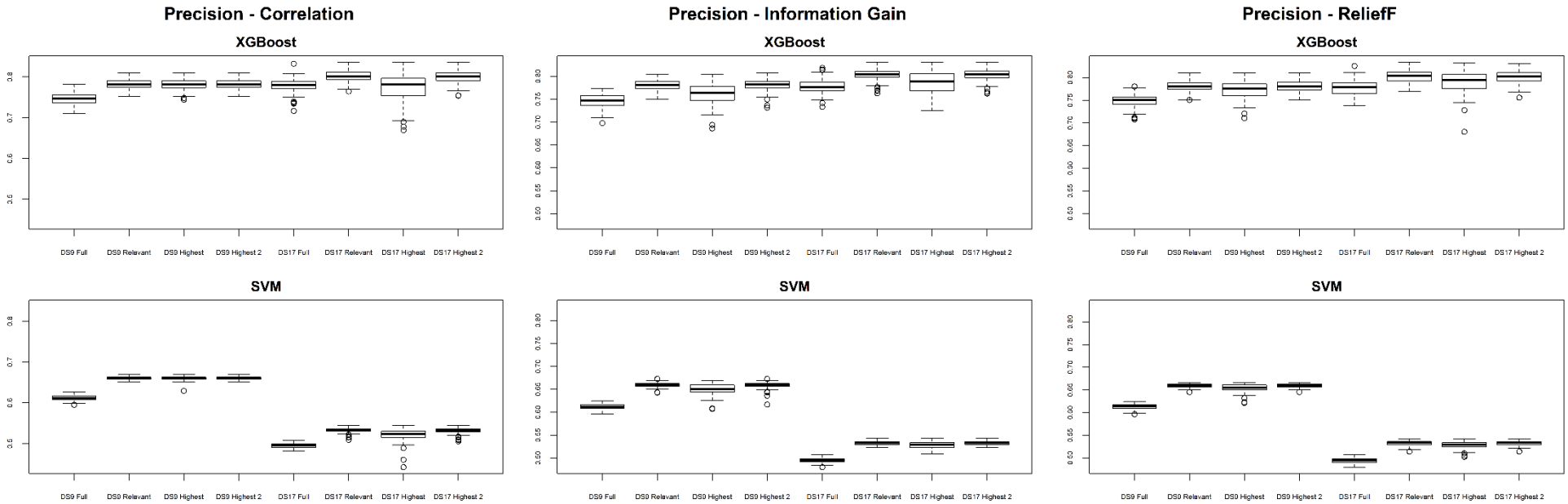


Figure H.31 Precision: Dataset 9 vs Dataset 17.

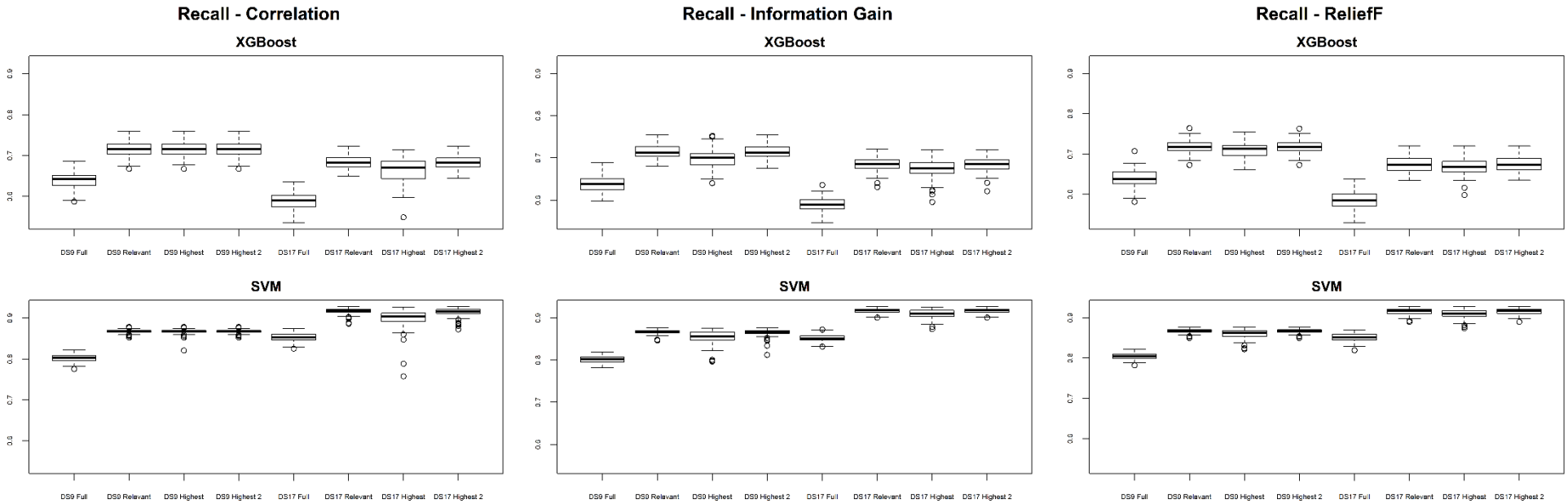


Figure H.32 Recall: Dataset 9 vs Dataset 17.

Table H.9: Structure of Dataset 16 and Dataset 24.

	k	p	q	ρ	Signal	Vector of Densities	Number of training instances	Number of test instances
Dataset 16	50	10	6	0.4	100	0.4	30	10 000
Dataset 24	50	10	6	0.4	100	vary	30	10 000

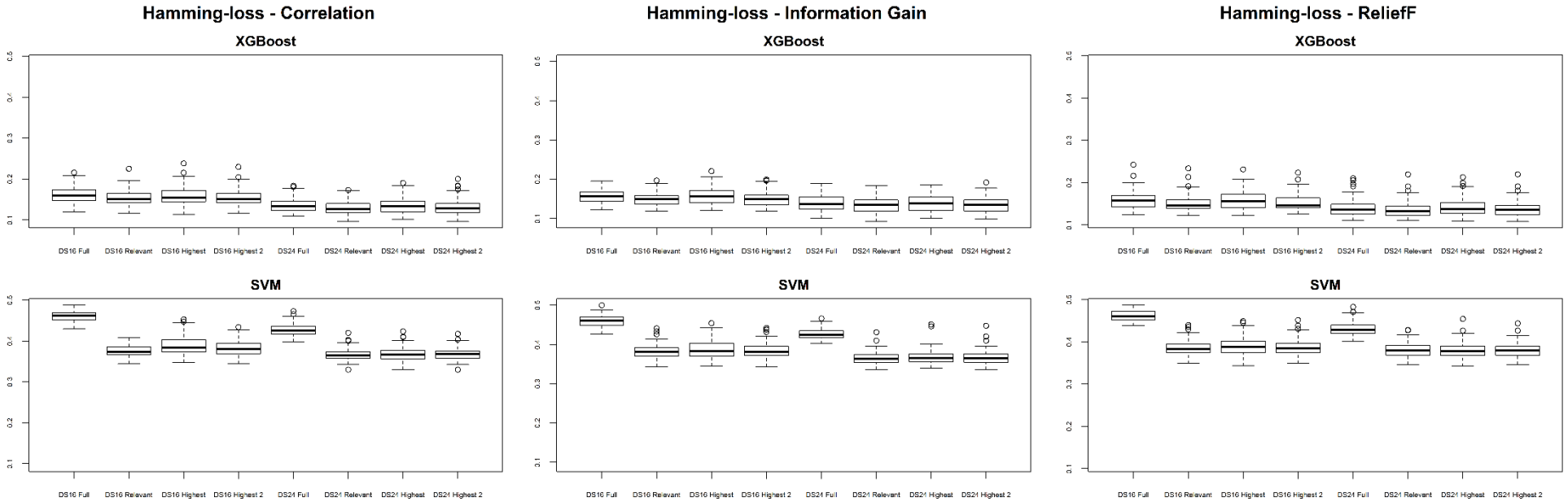


Figure H.33 Hamming-loss: Dataset 16 vs Dataset 24.

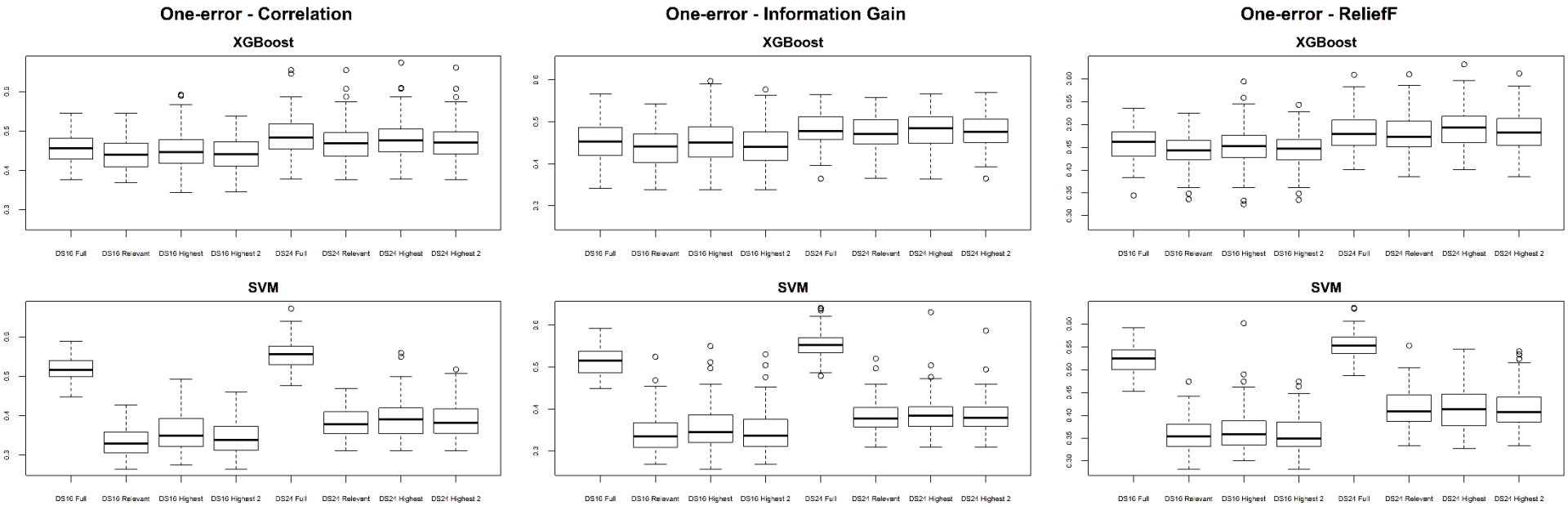


Figure H.34 One-error: Dataset 16 vs Dataset 24.

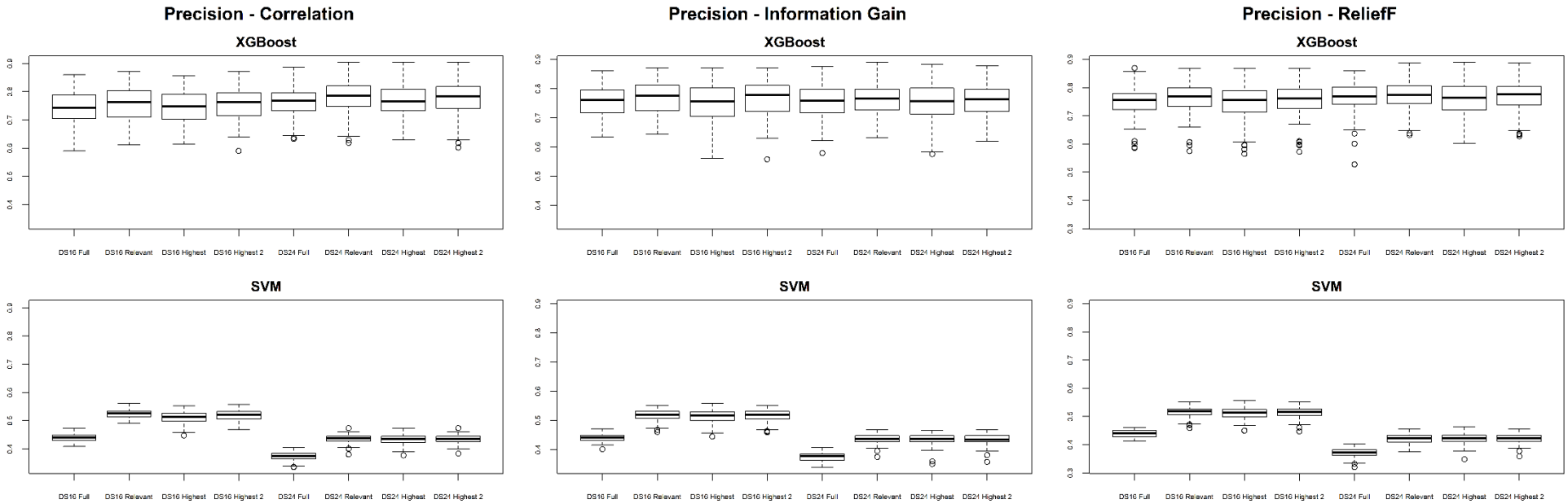


Figure H.35 Precision: Dataset 16 vs Dataset 24.

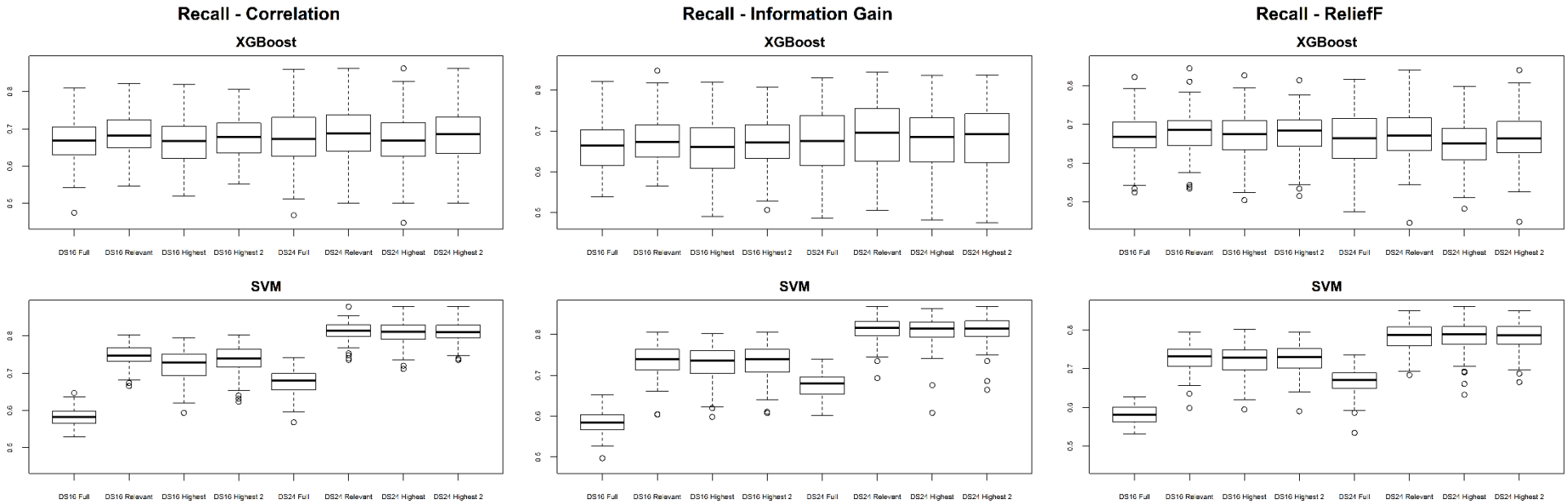


Figure H.36 Recall: Dataset 16 vs Dataset 24.

APPENDIX I

Table I.1: Four-way ANOVA for Dataset 1 vs Dataset 3.

Hamming-loss

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.006091	0.003045	18.11888	1.45E-08
Model	3	0.290548	0.096849	576.2337	0
Technique	1	4.943154	4.943154	29410.75	0
Dataset	1	20.38022	20.38022	121258.1	0
Measure:Model	6	0.001577	0.000263	1.563453	0.15354
Measure:Technique	2	0.000663	0.000332	1.972461	0.139228
Model:Technique	3	0.020936	0.006979	41.522	1.77E-26
Measure:Dataset	2	7.62E-05	3.81E-05	0.226654	0.797205
Model:Dataset	3	0.035177	0.011726	69.7644	3.8E-44
Technique:Dataset	1	1.594658	1.594658	9487.888	0
Measure:Model:Technique	6	0.000342	5.69E-05	0.338824	0.916604
Measure:Model:Dataset	6	0.004849	0.000808	4.808292	6.71E-05
Measure:Technique:Dataset	2	0.000182	9.11E-05	0.542028	0.581604
Model:Technique:Dataset	3	0.004036	0.001345	8.005134	2.55E-05
Measure:Model:Technique:Dataset	6	0.000144	2.41E-05	0.143179	0.990387
Residuals	4752	0.798683	0.000168		

One-error

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.018974	0.009487	17.14929	3.79E-08
Model	3	0.6325	0.210833	381.1154	8.2E-222
Technique	1	5.649046	5.649046	10211.56	0
Dataset	1	25.28774	25.28774	45711.69	0
Measure:Model	6	0.006572	0.001095	1.979961	0.064934
Measure:Technique	2	0.002361	0.00118	2.133701	0.118512
Model:Technique	3	0.026484	0.008828	15.95786	2.54E-10
Measure:Dataset	2	0.005331	0.002666	4.818415	0.008119
Model:Dataset	3	0.144892	0.048297	87.30539	5.48E-55
Technique:Dataset	1	0.047654	0.047654	86.14315	2.48E-20
Measure:Model:Technique	6	0.001195	0.000199	0.360116	0.904301
Measure:Model:Dataset	6	0.012032	0.002005	3.62491	0.001367
Measure:Technique:Dataset	2	0.001666	0.000833	1.505455	0.222022
Model:Technique:Dataset	3	0.003601	0.0012	2.17007	0.089409
Measure:Model:Technique:Dataset	6	0.000599	9.98E-05	0.18049	0.982258
Residuals	4752	2.62881	0.000553		

Precision

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010536	0.005268	13.75763	1.1E-06
Model	3	0.324615	0.108205	282.5758	8.8E-169
Technique	1	23.42165	23.42165	61165.35	0
Dataset	1	29.32038	29.32038	76569.8	0
Measure:Model	6	0.002681	0.000447	1.166833	0.320959
Measure:Technique	2	0.002735	0.001367	3.570676	0.028212
Model:Technique	3	0.024125	0.008042	21.00051	1.64E-13
Measure:Dataset	2	0.000536	0.000268	0.700029	0.496622
Model:Dataset	3	0.061571	0.020524	53.59748	4.6E-34
Technique:Dataset	1	4.682518	4.682518	12228.34	0
Measure:Model:Technique	6	0.000942	0.000157	0.409976	0.872889
Measure:Model:Dataset	6	0.007559	0.00126	3.28995	0.003122
Measure:Technique:Dataset	2	0.00176	0.00088	2.298511	0.10052
Model:Technique:Dataset	3	0.004288	0.001429	3.732355	0.010766
Measure:Model:Technique:Dataset	6	0.000709	0.000118	0.308722	0.932731
Residuals	4752	1.819653	0.000383		

Recall

Dataset 1 vs Dataset 3	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.01797	0.008985	14.18336	7.22E-07
Model	3	0.683852	0.227951	359.8359	1.4E-210
Technique	1	21.5256	21.5256	33979.63	0
Dataset	1	39.38182	39.38182	62166.9	0
Measure:Model	6	0.003513	0.000586	0.924253	0.476081
Measure:Technique	2	0.003331	0.001666	2.629373	0.072229
Model:Technique	3	0.004001	0.001334	2.105528	0.097334
Measure:Dataset	2	0.001332	0.000666	1.051629	0.34945
Model:Dataset	3	0.05921	0.019737	31.15563	6.15E-20
Technique:Dataset	1	4.595125	4.595125	7253.719	0
Measure:Model:Technique	6	0.000593	9.88E-05	0.15602	0.987908
Measure:Model:Dataset	6	0.008765	0.001461	2.306085	0.031686
Measure:Technique:Dataset	2	0.00036	0.00018	0.284042	0.752748
Model:Technique:Dataset	3	0.015292	0.005097	8.046648	2.4E-05
Measure:Model:Technique:Dataset	6	0.000266	4.44E-05	0.070031	0.998677
Residuals	4752	3.010323	0.000633		

Table I.2: Four-way ANOVA for Dataset 17 vs Dataset 21.

Hamming-loss

Dataset 17 vs Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000871	0.000436	13.39491	1.58E-06
Model	3	0.578591	0.192864	5929.748	0
Technique	1	31.11066	31.11066	956522.7	0
Dataset	1	8.739517	8.739517	268703.6	0
Measure:Model	6	0.001941	0.000323	9.946045	6.16E-11
Measure:Technique	2	4.9E-06	2.45E-06	0.07538	0.927392
Model:Technique	3	0.047085	0.015695	482.5575	1E-273
Measure:Dataset	2	0.000528	0.000264	8.117021	0.000303
Model:Dataset	3	0.036983	0.012328	379.021	1E-220
Technique:Dataset	1	1.857132	1.857132	57099.04	0
Measure:Model:Technique	6	4.62E-05	7.7E-06	0.236811	0.964553
Measure:Model:Dataset	6	0.002542	0.000424	13.02651	1.14E-14
Measure:Technique:Dataset	2	0.000154	7.68E-05	2.361091	0.094428
Model:Technique:Dataset	3	0.004235	0.001412	43.39831	1.16E-27
Measure:Model:Technique:Dataset	6	9.45E-05	1.58E-05	0.484265	0.820558
Residuals	4752	0.154558	3.25E-05		

One-error

Dataset 17 vs Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.003639	0.001819	13.47891	1.45E-06
Model	3	1.368211	0.45607	3378.919	0
Technique	1	1.075583	1.075583	7968.742	0
Dataset	1	12.25605	12.25605	90802.25	0
Measure:Model	6	0.008261	0.001377	10.20062	3.04E-11
Measure:Technique	2	0.000242	0.000121	0.894912	0.408712
Model:Technique	3	0.111237	0.037079	274.7098	2E-164
Measure:Dataset	2	0.005572	0.002786	20.64238	1.19E-09
Model:Dataset	3	0.378466	0.126155	934.655	0
Technique:Dataset	1	0.114892	0.114892	851.205	2.8E-172
Measure:Model:Technique	6	0.000154	2.57E-05	0.190686	0.979539
Measure:Model:Dataset	6	0.007901	0.001317	9.755923	1.04E-10
Measure:Technique:Dataset	2	0.000118	5.89E-05	0.436551	0.646288
Model:Technique:Dataset	3	0.013603	0.004534	33.59406	1.77E-21
Measure:Model:Technique:Dataset	6	0.000579	9.65E-05	0.71475	0.637715
Residuals	4752	0.641402	0.000135		

Precision

Dataset 17 vs Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004452	0.002226	18.60727	8.92E-09
Model	3	0.473906	0.157969	1320.494	0
Technique	1	136.0795	136.0795	1137519	0
Dataset	1	15.19762	15.19762	127040.3	0
Measure:Model	6	0.004906	0.000818	6.835114	3.1E-07
Measure:Technique	2	0.000976	0.000488	4.081134	0.016948
Model:Technique	3	0.099758	0.033253	277.9659	3.1E-166
Measure:Dataset	2	0.002584	0.001292	10.80012	2.09E-05
Model:Dataset	3	0.050331	0.016777	140.2434	4.54E-87
Technique:Dataset	1	5.33071	5.33071	44560.6	0
Measure:Model:Technique	6	0.000844	0.000141	1.176434	0.315669
Measure:Model:Dataset	6	0.006239	0.00104	8.692152	1.96E-09
Measure:Technique:Dataset	2	0.000337	0.000169	1.409744	0.244308
Model:Technique:Dataset	3	0.008476	0.002825	23.61642	3.62E-15
Measure:Model:Technique:Dataset	6	0.000834	0.000139	1.161732	0.323797
Residuals	4752	0.568474	0.00012		

Recall

Dataset 17 vs Dataset 21	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.002458	0.001229	6.499567	0.001517
Model	3	2.867824	0.955941	5055.756	0
Technique	1	24.54333	24.54333	129804.1	0
Dataset	1	33.47795	33.47795	177057.3	0
Measure:Model	6	0.006388	0.001065	5.630368	7.77E-06
Measure:Technique	2	0.001016	0.000508	2.687826	0.068132
Model:Technique	3	0.010619	0.00354	18.72125	4.54E-12
Measure:Dataset	2	0.004324	0.002162	11.43426	1.11E-05
Model:Dataset	3	0.342438	0.114146	603.6921	0
Technique:Dataset	1	11.90673	11.90673	62971.98	0
Measure:Model:Technique	6	0.00056	9.33E-05	0.493374	0.813779
Measure:Model:Dataset	6	0.006915	0.001152	6.095261	2.26E-06
Measure:Technique:Dataset	2	0.00402	0.00201	10.62928	2.48E-05
Model:Technique:Dataset	3	0.11349	0.03783	200.0747	3.4E-122
Measure:Model:Technique:Dataset	6	0.000707	0.000118	0.623634	0.711555
Residuals	4752	0.898507	0.000189		

Table I.3: Four-way ANOVA for Dataset 1 vs Dataset 9.

Hamming-loss

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.004043	0.002022	14.64135	4.58E-07
Model	3	0.763221	0.254407	1842.533	0
Technique	1	1.918268	1.918268	13892.97	0
Dataset	1	2.084919	2.084919	15099.94	0
Measure:Model	6	0.00779	0.001298	9.403007	2.77E-10
Measure:Technique	2	0.000146	7.32E-05	0.530067	0.588601
Model:Technique	3	0.019327	0.006442	46.65712	1.04E-29
Measure:Dataset	2	0.003866	0.001933	13.99957	8.67E-07
Model:Dataset	3	0.14868	0.04956	358.9353	4.1E-210
Technique:Dataset	1	0.180194	0.180194	1305.048	1E-252
Measure:Model:Technique	6	0.000519	8.64E-05	0.62592	0.709707
Measure:Model:Dataset	6	0.005614	0.000936	6.776995	3.62E-07
Measure:Technique:Dataset	2	0.000711	0.000356	2.57543	0.076227
Model:Technique:Dataset	3	0.003511	0.00117	8.476724	1.3E-05
Measure:Model:Technique:Dataset	6	0.000129	2.14E-05	0.155118	0.988093
Residuals	4752	0.656131	0.000138		

One-error

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.010621	0.00531	12.43001	4.13E-06
Model	3	1.985963	0.661988	1549.511	0
Technique	1	3.73064	3.73064	8732.292	0
Dataset	1	5.110752	5.110752	11962.72	0
Measure:Model	6	0.020273	0.003379	7.908719	1.68E-08
Measure:Technique	2	2.05E-05	1.02E-05	0.023962	0.976323
Model:Technique	3	0.078658	0.026219	61.37149	6.36E-39
Measure:Dataset	2	0.018182	0.009091	21.27911	6.31E-10
Model:Dataset	3	0.352089	0.117363	274.7113	2E-164
Technique:Dataset	1	0.440345	0.440345	1030.713	7.4E-205
Measure:Model:Technique	6	0.001611	0.000269	0.628668	0.707486
Measure:Model:Dataset	6	0.01686	0.00281	6.577228	6.2E-07
Measure:Technique:Dataset	2	0.001488	0.000744	1.741833	0.175311
Model:Technique:Dataset	3	0.023179	0.007726	18.08502	1.15E-11
Measure:Model:Technique:Dataset	6	0.000619	0.000103	0.241514	0.96278
Residuals	4752	2.030166	0.000427		

Precision

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.005467	0.002734	8.529125	0.000201
Model	3	0.786232	0.262077	817.7165	0
Technique	1	12.10614	12.10614	37772.78	0
Dataset	1	3.343844	3.343844	10433.24	0
Measure:Model	6	0.014767	0.002461	7.679187	3.14E-08
Measure:Technique	2	0.00127	0.000635	1.980761	0.138078
Model:Technique	3	0.029649	0.009883	30.83589	9.79E-20
Measure:Dataset	2	0.009343	0.004671	14.57544	4.89E-07
Model:Dataset	3	0.146041	0.04868	151.889	5.33E-94
Technique:Dataset	1	0.645948	0.645948	2015.442	0
Measure:Model:Technique	6	0.002919	0.000486	1.517839	0.167914
Measure:Model:Dataset	6	0.008379	0.001396	4.357247	0.000215
Measure:Technique:Dataset	2	0.003391	0.001696	5.290808	0.005067
Model:Technique:Dataset	3	0.005216	0.001739	5.42462	0.001009
Measure:Model:Technique:Dataset	6	0.000323	5.39E-05	0.168161	0.985258
Residuals	4752	1.523012	0.00032		

Recall

Dataset 1 vs Dataset 9	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.011318	0.005659	11.38104	1.17E-05
Model	3	2.196604	0.732201	1472.606	0
Technique	1	36.79046	36.79046	73993.08	0
Dataset	1	4.224117	4.224117	8495.555	0
Measure:Model	6	0.013507	0.002251	4.52769	0.000139
Measure:Technique	2	0.000322	0.000161	0.323711	0.723475
Model:Technique	3	0.015811	0.00527	10.59944	6.07E-07
Measure:Dataset	2	0.00676	0.00338	6.798138	0.001127
Model:Dataset	3	0.388926	0.129642	260.7361	1.2E-156
Technique:Dataset	1	0.515063	0.515063	1035.896	8.8E-206
Measure:Model:Technique	6	0.000893	0.000149	0.299324	0.937442
Measure:Model:Dataset	6	0.010276	0.001713	3.444508	0.002137
Measure:Technique:Dataset	2	0.001038	0.000519	1.043918	0.352153
Model:Technique:Dataset	3	0.00744	0.00248	4.987578	0.001868
Measure:Model:Technique:Dataset	6	0.000413	6.89E-05	0.138575	0.991196
Residuals	4752	2.362765	0.000497		

Table I.4: Four-way ANOVA for Dataset 5 vs Dataset 17.

Hamming-loss

Dataset 5 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.0098	0.0049	58.96156	5.08E-26
Model	3	0.481931	0.160644	1933.052	0
Technique	1	14.20226	14.20226	170898.1	0
Dataset	1	1.247632	1.247632	15012.96	0
Measure:Model	6	0.003331	0.000555	6.679638	4.71E-07
Measure:Technique	2	0.000233	0.000117	1.402279	0.246137
Model:Technique	3	0.013699	0.004566	54.94851	6.56E-35
Measure:Dataset	2	0.016252	0.008126	97.7807	2.43E-42
Model:Dataset	3	0.060343	0.020114	242.0382	3.7E-146
Technique:Dataset	1	0.199218	0.199218	2397.226	0
Measure:Model:Technique	6	5.62E-05	9.37E-06	0.112727	0.994982
Measure:Model:Dataset	6	0.009092	0.001515	18.2335	4.84E-21
Measure:Technique:Dataset	2	0.000668	0.000334	4.020678	0.018002
Model:Technique:Dataset	3	0.001573	0.000524	6.309111	0.000288
Measure:Model:Technique:Dataset	6	0.000196	3.27E-05	0.394014	0.883314
Residuals	4752	0.394909	8.31E-05		

One-error

Dataset 5 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.049542	0.024771	59.96704	1.9E-26
Model	3	2.010767	0.670256	1622.594	0
Technique	1	5.19418	5.19418	12574.37	0
Dataset	1	4.550128	4.550128	11015.21	0
Measure:Model	6	0.014909	0.002485	6.015518	2.79E-06
Measure:Technique	2	0.000391	0.000195	0.472803	0.623282
Model:Technique	3	0.117697	0.039232	94.97619	1.07E-59
Measure:Dataset	2	0.085139	0.042569	103.0544	1.54E-44
Model:Dataset	3	0.1313	0.043767	105.9528	2.13E-66
Technique:Dataset	1	0.815436	0.815436	1974.054	0
Measure:Model:Technique	6	0.000205	3.42E-05	0.082689	0.997883
Measure:Model:Dataset	6	0.034599	0.005767	13.95997	8.3E-16
Measure:Technique:Dataset	2	0.001092	0.000546	1.321778	0.266759
Model:Technique:Dataset	3	0.01225	0.004083	9.885511	1.7E-06
Measure:Model:Technique:Dataset	6	0.000902	0.00015	0.363991	0.901986
Residuals	4752	1.96294	0.000413		

Precision

Dataset 5 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.009559	0.004779	11.45379	1.09E-05
Model	3	0.509076	0.169692	406.6688	3.9E-235
Technique	1	67.0371	67.0371	160655.3	0
Dataset	1	3.378242	3.378242	8096.001	0
Measure:Model	6	0.006204	0.001034	2.478135	0.021441
Measure:Technique	2	0.000658	0.000329	0.787874	0.45487
Model:Technique	3	0.029263	0.009754	23.37607	5.14E-15
Measure:Dataset	2	0.035111	0.017555	42.07192	7.73E-19
Model:Dataset	3	0.029633	0.009878	23.67229	3.34E-15
Technique:Dataset	1	1.366237	1.366237	3274.203	0
Measure:Model:Technique	6	0.000697	0.000116	0.278209	0.947439
Measure:Model:Dataset	6	0.025574	0.004262	10.21459	2.92E-11
Measure:Technique:Dataset	2	0.000844	0.000422	1.010833	0.363994
Model:Technique:Dataset	3	0.01902	0.00634	15.19415	7.72E-10
Measure:Model:Technique:Dataset	6	0.004639	0.000773	1.853074	0.085032
Residuals	4752	1.982881	0.000417		

Recall

Dataset 5 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.060763	0.030382	54.00413	6.44E-24
Model	3	2.682627	0.894209	1589.473	0
Technique	1	89.3578	89.3578	158835.2	0
Dataset	1	4.43649	4.43649	7885.943	0
Measure:Model	6	0.016017	0.00267	4.745141	7.91E-05
Measure:Technique	2	0.005362	0.002681	4.765313	0.008561
Model:Technique	3	0.071708	0.023903	42.48745	4.35E-27
Measure:Dataset	2	0.05582	0.02791	49.61039	4.75E-22
Model:Dataset	3	0.401724	0.133908	238.0237	6.9E-144
Technique:Dataset	1	1.098703	1.098703	1952.964	0
Measure:Model:Technique	6	0.000715	0.000119	0.211901	0.973192
Measure:Model:Dataset	6	0.021977	0.003663	6.51073	7.42E-07
Measure:Technique:Dataset	2	0.00793	0.003965	7.04771	0.000879
Model:Technique:Dataset	3	0.029851	0.00995	17.68689	2.05E-11
Measure:Model:Technique:Dataset	6	0.000506	8.43E-05	0.149777	0.989154
Residuals	4752	2.67339	0.000563		

Table I.5: Four-way ANOVA for Dataset 1 vs Dataset 2.

Hamming-loss

Dataset 1 vs Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.009497	0.004748	26.69657	2.95E-12
Model	3	0.297049	0.099016	556.7023	0
Technique	1	13.99836	13.99836	78703.36	0
Dataset	1	2.387459	2.387459	13423.08	0
Measure:Model	6	0.004389	0.000731	4.112655	0.000401
Measure:Technique	2	0.000769	0.000384	2.160992	0.115324
Model:Technique	3	0.023908	0.007969	44.80664	1.51E-28
Measure:Dataset	2	0.00041	0.000205	1.152611	0.315899
Model:Dataset	3	0.008403	0.002801	15.74727	3.45E-10
Technique:Dataset	1	7.73349	7.73349	43480.22	0
Measure:Model:Technique	6	0.000331	5.51E-05	0.309961	0.932098
Measure:Model:Dataset	6	0.006273	0.001045	5.877794	4.03E-06
Measure:Technique:Dataset	2	0.001405	0.000703	3.949943	0.019319
Model:Technique:Dataset	3	0.005513	0.001838	10.33112	8.94E-07
Measure:Model:Technique:Dataset	6	0.000203	3.38E-05	0.18978	0.979789
Residuals	4752	0.845202	0.000178		

One-error

Dataset 1 vs Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.029021	0.01451	24.46724	2.68E-11
Model	3	1.001617	0.333872	562.9737	0
Technique	1	5.321866	5.321866	8973.702	0
Dataset	1	57.55878	57.55878	97055.31	0
Measure:Model	6	0.014937	0.00249	4.197815	0.000323
Measure:Technique	2	0.003718	0.001859	3.134429	0.043615
Model:Technique	3	0.0547	0.018233	30.74472	1.12E-19
Measure:Dataset	2	0.001803	0.000901	1.520102	0.218796
Model:Dataset	3	0.029451	0.009817	16.55311	1.07E-10
Technique:Dataset	1	0.083033	0.083033	140.0094	7.39E-32
Measure:Model:Technique	6	0.002093	0.000349	0.588321	0.739961
Measure:Model:Dataset	6	0.018161	0.003027	5.103778	3.11E-05
Measure:Technique:Dataset	2	0.007433	0.003717	6.26681	0.001914
Model:Technique:Dataset	3	0.006471	0.002157	3.637257	0.012277
Measure:Model:Technique:Dataset	6	0.001754	0.000292	0.493011	0.814051
Residuals	4752	2.81818	0.000593		

Precision

Dataset 1 vs Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.021827	0.010913	19.97525	2.3E-09
Model	3	0.418587	0.139529	255.3852	1.2E-153
Technique	1	16.94616	16.94616	31017.2	0
Dataset	1	8.289503	8.289503	15172.59	0
Measure:Model	6	0.009322	0.001554	2.843615	0.009144
Measure:Technique	2	0.003135	0.001568	2.869086	0.056849
Model:Technique	3	0.026504	0.008835	16.17019	1.87E-10
Measure:Dataset	2	0.001166	0.000583	1.066698	0.344225
Model:Dataset	3	0.007859	0.00262	4.795107	0.002448
Technique:Dataset	1	2.07617	2.07617	3800.092	0
Measure:Model:Technique	6	0.001583	0.000264	0.482875	0.821587
Measure:Model:Dataset	6	0.011398	0.0019	3.476948	0.001972
Measure:Technique:Dataset	2	0.001392	0.000696	1.27354	0.279935
Model:Technique:Dataset	3	0.00238	0.000793	1.452364	0.225542
Measure:Model:Technique:Dataset	6	0.00094	0.000157	0.286616	0.943558
Residuals	4752	2.596242	0.000546		

Recall

Dataset 1 vs Dataset 2	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.015999	0.008	9.085963	0.000115
Model	3	0.889888	0.296629	336.9156	2.3E-198
Technique	1	28.94241	28.94241	32873.18	0
Dataset	1	8.707447	8.707447	9890.035	0
Measure:Model	6	0.012121	0.00202	2.294581	0.032516
Measure:Technique	2	0.008374	0.004187	4.755778	0.008643
Model:Technique	3	0.029837	0.009946	11.2964	2.21E-07
Measure:Dataset	2	0.000167	8.37E-05	0.095034	0.909344
Model:Dataset	3	0.004801	0.0016	1.817559	0.141652
Technique:Dataset	1	1.969469	1.969469	2236.949	0
Measure:Model:Technique	6	0.000889	0.000148	0.168319	0.985222
Measure:Model:Dataset	6	0.011312	0.001885	2.141398	0.045708
Measure:Technique:Dataset	2	0.0181	0.00905	10.27891	3.51E-05
Model:Technique:Dataset	3	0.030214	0.010071	11.4392	1.8E-07
Measure:Model:Technique:Dataset	6	0.001197	0.0002	0.226614	0.968243
Residuals	4752	4.183786	0.00088		

Table I.6: Four-way ANOVA for Dataset 10 vs Dataset 12.

Hamming-loss

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.143065	0.071532	148.1522	3.84E-63
Model	3	0.412797	0.137599	284.984	4.1E-170
Technique	1	18.31563	18.31563	37933.83	0
Dataset	1	0.370693	0.370693	767.7496	8.9E-157
Measure:Model	6	0.050977	0.008496	17.59655	2.93E-20
Measure:Technique	2	0.020963	0.010482	21.70886	4.12E-10
Model:Technique	3	0.170698	0.056899	117.845	1.3E-73
Measure:Dataset	2	0.061872	0.030936	64.07238	3.49E-28
Model:Dataset	3	0.017065	0.005688	11.78118	1.1E-07
Technique:Dataset	1	6.038925	6.038925	12507.32	0
Measure:Model:Technique	6	0.010222	0.001704	3.528549	0.001736
Measure:Model:Dataset	6	0.034998	0.005833	12.08094	1.61E-13
Measure:Technique:Dataset	2	0.020569	0.010284	21.29996	6.18E-10
Model:Technique:Dataset	3	0.014485	0.004828	9.999921	1.44E-06
Measure:Model:Technique:Dataset	6	0.005625	0.000938	1.941673	0.070482
Residuals	4752	2.294413	0.000483		

One-error

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.416328	0.208164	151.0317	2.56E-64
Model	3	1.438868	0.479623	347.9859	2.8E-204
Technique	1	0.411713	0.411713	298.7144	5.86E-65
Dataset	1	9.861025	9.861025	7154.579	0
Measure:Model	6	0.155905	0.025984	18.85257	8.4E-22
Measure:Technique	2	0.155252	0.077626	56.32076	6.69E-25
Model:Technique	3	0.976144	0.325381	236.0776	8.8E-143
Measure:Dataset	2	0.25659	0.128295	93.08327	2.22E-40
Model:Dataset	3	0.13044	0.04348	31.54655	3.48E-20
Technique:Dataset	1	0.021219	0.021219	15.39515	8.84E-05
Measure:Model:Technique	6	0.032188	0.005365	3.892345	0.0007
Measure:Model:Dataset	6	0.101975	0.016996	12.33116	8.01E-14
Measure:Technique:Dataset	2	0.071644	0.035822	25.99035	5.94E-12
Model:Technique:Dataset	3	0.066243	0.022081	16.02074	2.32E-10
Measure:Model:Technique:Dataset	6	0.018907	0.003151	2.286318	0.033124
Residuals	4752	6.549594	0.001378		

Precision

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.157047	0.078523	60.53223	1.1E-26
Model	3	0.505837	0.168612	129.9802	6.34E-81
Technique	1	11.0048	11.0048	8483.408	0
Dataset	1	0.095317	0.095317	73.47813	1.36E-17
Measure:Model	6	0.070497	0.01175	9.057521	7.18E-10
Measure:Technique	2	0.013557	0.006778	5.225336	0.00541
Model:Technique	3	0.120712	0.040237	31.01823	7.51E-20
Measure:Dataset	2	0.101606	0.050803	39.16325	1.35E-17
Model:Dataset	3	0.009553	0.003184	2.45466	0.061293
Technique:Dataset	1	2.362238	2.362238	1821.008	0
Measure:Model:Technique	6	0.006969	0.001161	0.895351	0.497143
Measure:Model:Dataset	6	0.046143	0.007691	5.928525	3.52E-06
Measure:Technique:Dataset	2	0.006075	0.003038	2.341733	0.096272
Model:Technique:Dataset	3	0.024908	0.008303	6.400372	0.000253
Measure:Model:Technique:Dataset	6	0.004509	0.000751	0.579265	0.747189
Residuals	4752	6.164363	0.001297		

Recall

Dataset 10 vs Dataset 12	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.262174	0.131087	57.42525	2.28E-25
Model	3	1.025521	0.34184	149.75	9.91E-93
Technique	1	16.8208	16.8208	7368.686	0
Dataset	1	0.218247	0.218247	95.60736	2.27E-22
Measure:Model	6	0.123706	0.020618	9.03199	7.7E-10
Measure:Technique	2	0.09881	0.049405	21.64276	4.4E-10
Model:Technique	3	0.562894	0.187631	82.19572	7.69E-52
Measure:Dataset	2	0.145918	0.072959	31.96111	1.63E-14
Model:Dataset	3	0.019922	0.006641	2.909027	0.033255
Technique:Dataset	1	2.197473	2.197473	962.647	1.3E-192
Measure:Model:Technique	6	0.021287	0.003548	1.554171	0.156375
Measure:Model:Dataset	6	0.074272	0.012379	5.422746	1.34E-05
Measure:Technique:Dataset	2	0.091433	0.045716	20.02701	2.18E-09
Model:Technique:Dataset	3	0.002437	0.000812	0.355837	0.784924
Measure:Model:Technique:Dataset	6	0.006552	0.001092	0.478406	0.824882
Residuals	4752	10.84758	0.002283		

Table I.7: Four-way ANOVA for Dataset 17 vs Dataset 19.

Hamming-loss

Dataset 17 vs Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.001909	0.000954	22.02671	3.01E-10
Model	3	0.665815	0.221938	5121.991	0
Technique	1	31.70848	31.70848	731782.2	0
Dataset	1	3.312986	3.312986	76458.53	0
Measure:Model	6	0.000782	0.00013	3.008456	0.006172
Measure:Technique	2	0.00011	5.52E-05	1.274918	0.279549
Model:Technique	3	0.028646	0.009549	220.369	7.5E-134
Measure:Dataset	2	0.003815	0.001907	44.01682	1.14E-19
Model:Dataset	3	0.015405	0.005135	118.5102	5.15E-74
Technique:Dataset	1	2.005345	2.005345	46280.21	0
Measure:Model:Technique	6	0.000107	1.78E-05	0.411679	0.871757
Measure:Model:Dataset	6	0.008633	0.001439	33.20659	1.94E-39
Measure:Technique:Dataset	2	0.000218	0.000109	2.510756	0.081315
Model:Technique:Dataset	3	0.000228	7.6E-05	1.754934	0.153558
Measure:Model:Technique:Dataset	6	0.000144	2.39E-05	0.552428	0.768423
Residuals	4752	0.205907	4.33E-05		

One-error

Dataset 17 vs Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.0069	0.00345	14.97123	3.3E-07
Model	3	3.353354	1.117785	4850.946	0
Technique	1	2.836233	2.836233	12308.65	0
Dataset	1	59.82884	59.82884	259644.4	0
Measure:Model	6	0.003821	0.000637	2.763575	0.011046
Measure:Technique	2	0.000612	0.000306	1.326952	0.265383
Model:Technique	3	0.172219	0.057406	249.131	3.8E-150
Measure:Dataset	2	0.022534	0.011267	48.89541	9.56E-22
Model:Dataset	3	0.012415	0.004138	17.95994	1.38E-11
Technique:Dataset	1	0.094896	0.094896	411.8297	7.06E-88
Measure:Model:Technique	6	0.000706	0.000118	0.510456	0.800894
Measure:Model:Dataset	6	0.028128	0.004688	20.34504	1.23E-23
Measure:Technique:Dataset	2	0.0013	0.00065	2.820936	0.05965
Model:Technique:Dataset	3	0.001527	0.000509	2.208967	0.084937
Measure:Model:Technique:Dataset	6	0.000458	7.63E-05	0.330929	0.920982
Residuals	4752	1.094985	0.00023		

Precision

Dataset 17 vs Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.003184	0.001592	6.783864	0.001143
Model	3	0.69515	0.231717	987.2729	0
Technique	1	87.98243	87.98243	374865.8	0
Dataset	1	10.29681	10.29681	43871.49	0
Measure:Model	6	0.002113	0.000352	1.500573	0.173651
Measure:Technique	2	0.000848	0.000424	1.806958	0.164265
Model:Technique	3	0.058277	0.019426	82.7665	3.42E-52
Measure:Dataset	2	0.023728	0.011864	50.54825	1.89E-22
Model:Dataset	3	0.003423	0.001141	4.861829	0.002229
Technique:Dataset	1	0.000548	0.000548	2.336031	0.126478
Measure:Model:Technique	6	0.000622	0.000104	0.44169	0.851257
Measure:Model:Dataset	6	0.024969	0.004161	17.73077	2E-20
Measure:Technique:Dataset	2	0.006054	0.003027	12.89776	2.59E-06
Model:Technique:Dataset	3	0.005055	0.001685	7.178805	8.33E-05
Measure:Model:Technique:Dataset	6	0.005659	0.000943	4.018294	0.000509
Residuals	4752	1.115313	0.000235		

Recall

Dataset 17 vs Dataset 19	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.014256	0.007128	20.56603	1.28E-09
Model	3	4.615494	1.538498	4439.058	0
Technique	1	66.66052	66.66052	192336.9	0
Dataset	1	12.04085	12.04085	34741.69	0
Measure:Model	6	0.004666	0.000778	2.243728	0.036435
Measure:Technique	2	2.97E-05	1.48E-05	0.042818	0.958086
Model:Technique	3	0.081828	0.027276	78.69952	1.11E-49
Measure:Dataset	2	0.003792	0.001896	5.471216	0.004233
Model:Dataset	3	0.015705	0.005235	15.10465	8.79E-10
Technique:Dataset	1	0.05767	0.05767	166.397	1.91E-37
Measure:Model:Technique	6	0.000991	0.000165	0.476506	0.826278
Measure:Model:Dataset	6	0.026338	0.00439	12.66549	3.14E-14
Measure:Technique:Dataset	2	0.007753	0.003876	11.18455	1.43E-05
Model:Technique:Dataset	3	0.023595	0.007865	22.69322	1.39E-14
Measure:Model:Technique:Dataset	6	0.000344	5.73E-05	0.165437	0.985879
Residuals	4752	1.646958	0.000347		

Table I.8: Four-way ANOVA for Dataset 9 vs Dataset 17.

Hamming-loss

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000714	0.000357	7.279461	0.000697
Model	3	1.133551	0.37785	7706.622	0
Technique	1	9.073454	9.073454	185061.9	0
Dataset	1	0.073936	0.073936	1507.999	9.3E-287
Measure:Model	6	0.00178	0.000297	6.051384	2.54E-06
Measure:Technique	2	4.34E-05	2.17E-05	0.44224	0.642622
Model:Technique	3	0.028766	0.009589	195.5682	1.4E-119
Measure:Dataset	2	0.005509	0.002755	56.18114	7.67E-25
Model:Dataset	3	0.018351	0.006117	124.7625	8.71E-78
Technique:Dataset	1	1.44651	1.44651	29502.97	0
Measure:Model:Technique	6	0.000199	3.32E-05	0.67637	0.668801
Measure:Model:Dataset	6	0.011386	0.001898	38.70453	3.64E-46
Measure:Technique:Dataset	2	0.000156	7.79E-05	1.588188	0.204404
Model:Technique:Dataset	3	0.00028	9.33E-05	1.902591	0.126882
Measure:Model:Technique:Dataset	6	8.72E-05	1.45E-05	0.296562	0.938797
Residuals	4752	0.232987	4.9E-05		

One-error

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000293	0.000146	0.872693	0.417892
Model	3	3.297327	1.099109	6552.396	0
Technique	1	1.747633	1.747633	10418.6	0
Dataset	1	4.624407	4.624407	27568.65	0
Measure:Model	6	0.003767	0.000628	3.742778	0.001019
Measure:Technique	2	0.000236	0.000118	0.704224	0.494544
Model:Technique	3	0.189414	0.063138	376.4005	2.5E-219
Measure:Dataset	2	0.019481	0.009741	58.06907	1.21E-25
Model:Dataset	3	0.009719	0.00324	19.31328	1.92E-12
Technique:Dataset	1	0.002925	0.002925	17.43476	3.03E-05
Measure:Model:Technique	6	0.001041	0.000173	1.034176	0.400772
Measure:Model:Dataset	6	0.032308	0.005385	32.10137	4.39E-38
Measure:Technique:Dataset	2	0.000903	0.000452	2.692093	0.067842
Model:Technique:Dataset	3	0.000264	8.81E-05	0.525355	0.664851
Measure:Model:Technique:Dataset	6	0.000119	1.99E-05	0.118552	0.994238
Residuals	4752	0.797108	0.000168		

Precision

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.002124	0.001062	6.424337	0.001636
Model	3	1.022735	0.340912	2062.225	0
Technique	1	46.50951	46.50951	281342.8	0
Dataset	1	3.243593	3.243593	19620.97	0
Measure:Model	6	0.003651	0.000608	3.680662	0.00119
Measure:Technique	2	0.000693	0.000347	2.096497	0.123
Model:Technique	3	0.071648	0.023883	144.4698	1.37E-89
Measure:Dataset	2	0.011686	0.005843	35.34453	5.8E-16
Model:Dataset	3	0.031868	0.010623	64.25876	1.01E-40
Technique:Dataset	1	6.434804	6.434804	38925.06	0
Measure:Model:Technique	6	0.001067	0.000178	1.075351	0.374665
Measure:Model:Dataset	6	0.022287	0.003715	22.46961	3E-26
Measure:Technique:Dataset	2	0.000847	0.000424	2.562804	0.077195
Model:Technique:Dataset	3	0.002901	0.000967	5.849215	0.000553
Measure:Model:Technique:Dataset	6	0.002364	0.000394	2.383701	0.026592
Residuals	4752	0.785565	0.000165		

Recall

Dataset 9 vs Dataset 17	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.000652	0.000326	1.399751	0.24676
Model	3	4.652344	1.550781	6659.511	0
Technique	1	47.2833	47.2833	203048.4	0
Dataset	1	0.020295	0.020295	87.15163	1.51E-20
Measure:Model	6	0.003128	0.000521	2.238621	0.036852
Measure:Technique	2	0.001144	0.000572	2.456713	0.085825
Model:Technique	3	0.094066	0.031355	134.6486	1.01E-83
Measure:Dataset	2	0.015271	0.007636	32.79014	7.19E-15
Model:Dataset	3	0.015629	0.00521	22.3718	2.22E-14
Technique:Dataset	1	2.33616	2.33616	10032.16	0
Measure:Model:Technique	6	0.00037	6.17E-05	0.264787	0.953356
Measure:Model:Dataset	6	0.02481	0.004135	17.75693	1.86E-20
Measure:Technique:Dataset	2	0.003661	0.00183	7.860066	0.000391
Model:Technique:Dataset	3	0.017578	0.005859	25.1618	3.81E-16
Measure:Model:Technique:Dataset	6	0.001039	0.000173	0.743456	0.614612
Residuals	4752	1.106585	0.000233		

Table I.9: Four-way ANOVA for Dataset 16 vs Dataset 24.

Hamming-loss

Dataset 16 vs Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.021184	0.010592	31.4995	2.57E-14
Model	3	1.123915	0.374638	1114.149	0
Technique	1	74.48286	74.48286	221507.1	0
Dataset	1	0.409823	0.409823	1218.785	6.5E-238
Measure:Model	6	0.004063	0.000677	2.013869	0.060361
Measure:Technique	2	0.009766	0.004883	14.52208	5.16E-07
Model:Technique	3	0.879166	0.293055	871.5273	0
Measure:Dataset	2	0.010016	0.005008	14.89384	3.56E-07
Model:Dataset	3	0.028315	0.009438	28.06913	5.5E-18
Technique:Dataset	1	0.000137	0.000137	0.40866	0.522681
Measure:Model:Technique	6	0.002867	0.000478	1.421133	0.202251
Measure:Model:Dataset	6	0.002028	0.000338	1.00503	0.419961
Measure:Technique:Dataset	2	0.004989	0.002495	7.418606	0.000607
Model:Technique:Dataset	3	0.017157	0.005719	17.0076	5.51E-11
Measure:Model:Technique:Dataset	6	0.001189	0.000198	0.589214	0.739247
Residuals	4752	1.597884	0.000336		

One-error

Dataset 16 vs Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.124813	0.062406	33.98414	2.22E-15
Model	3	6.548669	2.18289	1188.718	0
Technique	1	3.016412	3.016412	1642.623	0
Dataset	1	1.660273	1.660273	904.1214	5.4E-182
Measure:Model	6	0.038443	0.006407	3.4891	0.001914
Measure:Technique	2	0.058431	0.029216	15.90976	1.3E-07
Model:Technique	3	5.389589	1.79653	978.3216	0
Measure:Dataset	2	0.010174	0.005087	2.770081	0.062758
Model:Dataset	3	0.01607	0.005357	2.916986	0.032898
Technique:Dataset	1	0.022792	0.022792	12.41184	0.000431
Measure:Model:Technique	6	0.020599	0.003433	1.869554	0.082133
Measure:Model:Dataset	6	0.015608	0.002601	1.416629	0.203985
Measure:Technique:Dataset	2	0.005516	0.002758	1.501978	0.222795
Model:Technique:Dataset	3	0.005795	0.001932	1.051934	0.368326
Measure:Model:Technique:Dataset	6	0.004454	0.000742	0.404221	0.876685
Residuals	4752	8.726282	0.001836		

Precision

Dataset 16 vs Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.023677	0.011839	6.412657	0.001655
Model	3	1.223752	0.407917	220.9564	3.5E-134
Technique	1	110.6785	110.6785	59951.17	0
Dataset	1	1.466578	1.466578	794.4002	9.4E-162
Measure:Model	6	0.007544	0.001257	0.681091	0.66497
Measure:Technique	2	0.011604	0.005802	3.142869	0.043249
Model:Technique	3	0.826541	0.275514	149.2374	2E-92
Measure:Dataset	2	0.023775	0.011888	6.43911	0.001612
Model:Dataset	3	0.032781	0.010927	5.918858	0.000501
Technique:Dataset	1	2.480678	2.480678	1343.707	2.7E-259
Measure:Model:Technique	6	0.004316	0.000719	0.389623	0.886124
Measure:Model:Dataset	6	0.001392	0.000232	0.125695	0.99324
Measure:Technique:Dataset	2	0.03789	0.018945	10.26185	3.57E-05
Model:Technique:Dataset	3	0.014454	0.004818	2.609838	0.049795
Measure:Model:Technique:Dataset	6	0.003625	0.000604	0.327245	0.92299
Residuals	4752	8.772879	0.001846		

Recall

Dataset 16 vs Dataset 24	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Measure	2	0.124245	0.062122	21.79503	3.78E-10
Model	3	4.789772	1.596591	560.1491	0
Technique	1	3.973393	3.973393	1394.028	8.8E-268
Dataset	1	1.848486	1.848486	648.5243	3.3E-134
Measure:Model	6	0.016245	0.002707	0.949888	0.457819
Measure:Technique	2	0.01573	0.007865	2.759378	0.063433
Model:Technique	3	4.026728	1.342243	470.9134	6.9E-268
Measure:Dataset	2	0.086268	0.043134	15.13312	2.81E-07
Model:Dataset	3	0.025724	0.008575	3.008355	0.02906
Technique:Dataset	1	1.604377	1.604377	562.8808	1.1E-117
Measure:Model:Technique	6	0.011643	0.001941	0.680832	0.66518
Measure:Model:Dataset	6	0.009435	0.001572	0.551689	0.769004
Measure:Technique:Dataset	2	0.001249	0.000624	0.219022	0.803312
Model:Technique:Dataset	3	0.022871	0.007624	2.674747	0.045635
Measure:Model:Technique:Dataset	6	0.005091	0.000848	0.297688	0.938246
Residuals	4752	13.54461	0.00285		

APPENDIX J

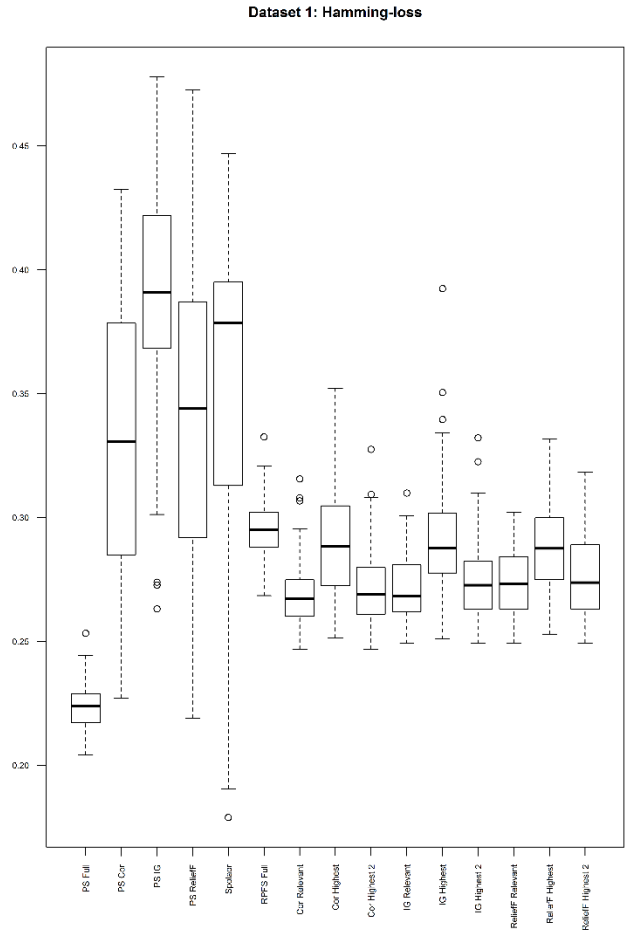


Figure J.1 Dataset 1: Hamming-loss.

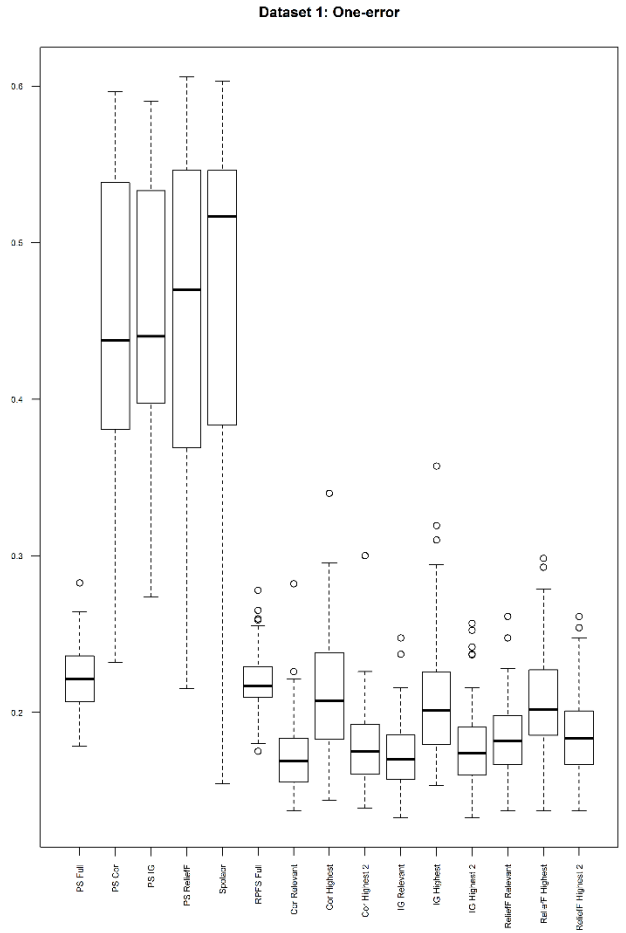


Figure J.2 Dataset 1: One-error.

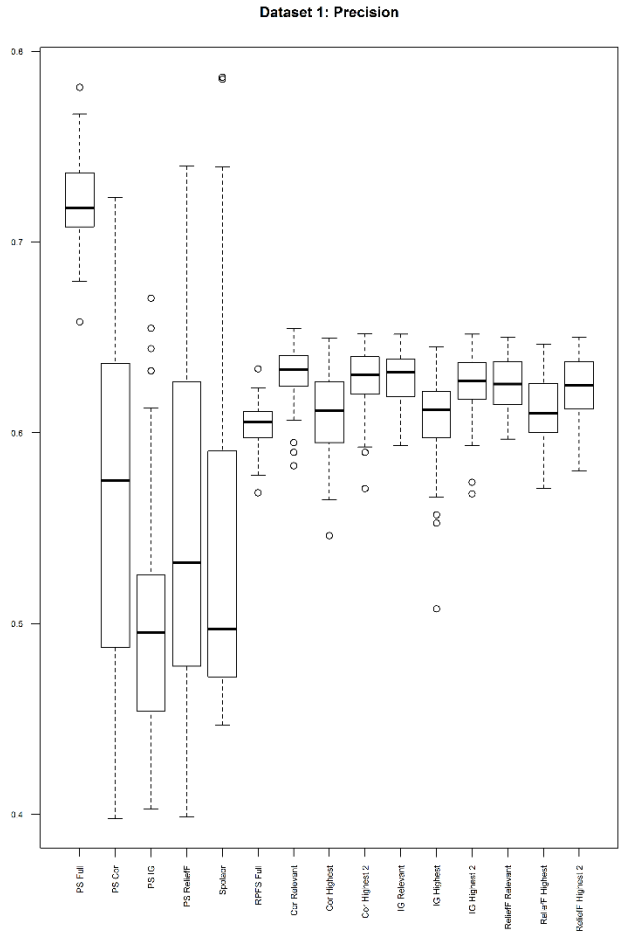


Figure J.3 Dataset 1: Precision.

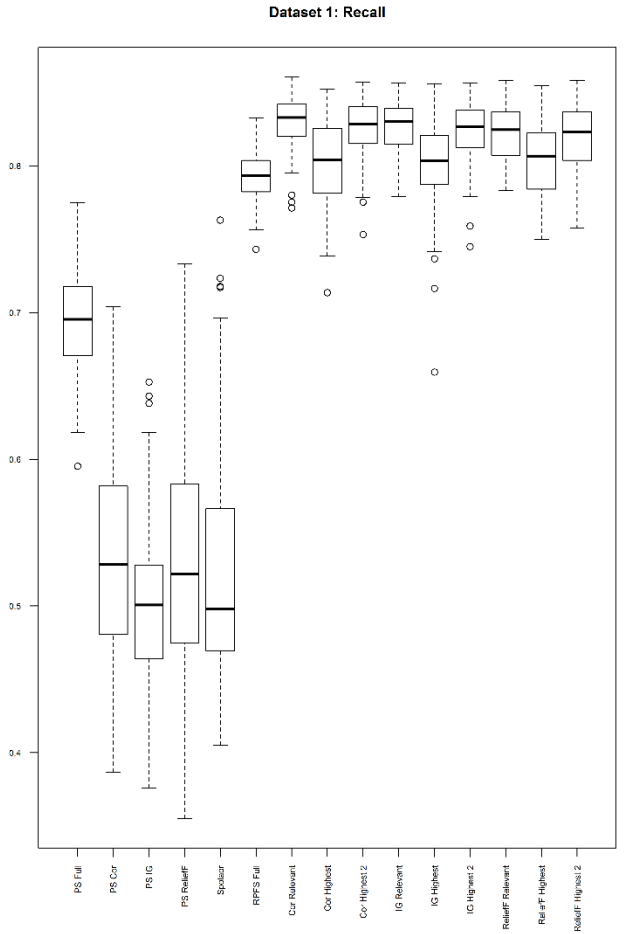


Figure J.4 Dataset 1: Recall.

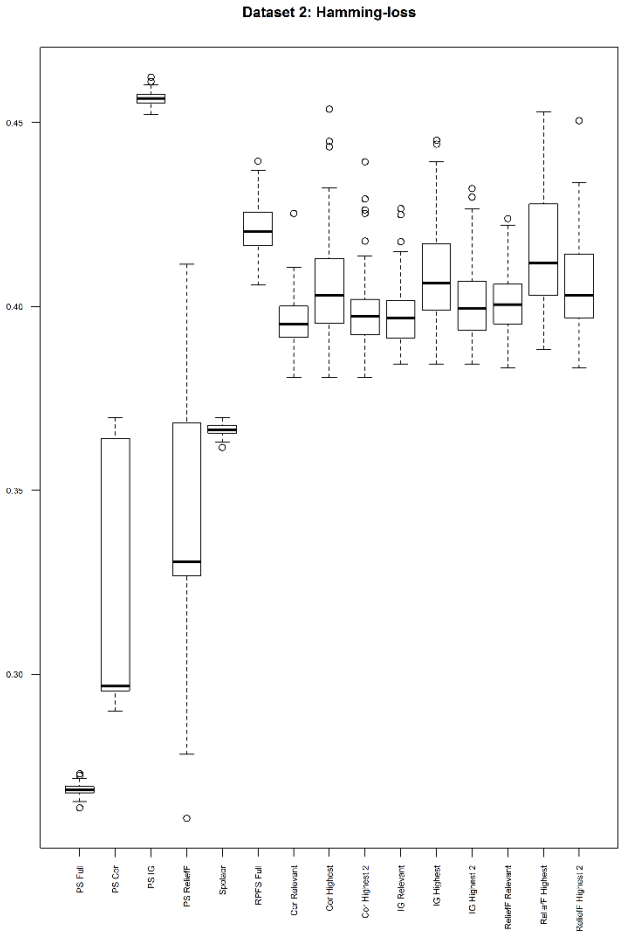


Figure J.5 Dataset 2: Hamming-loss.

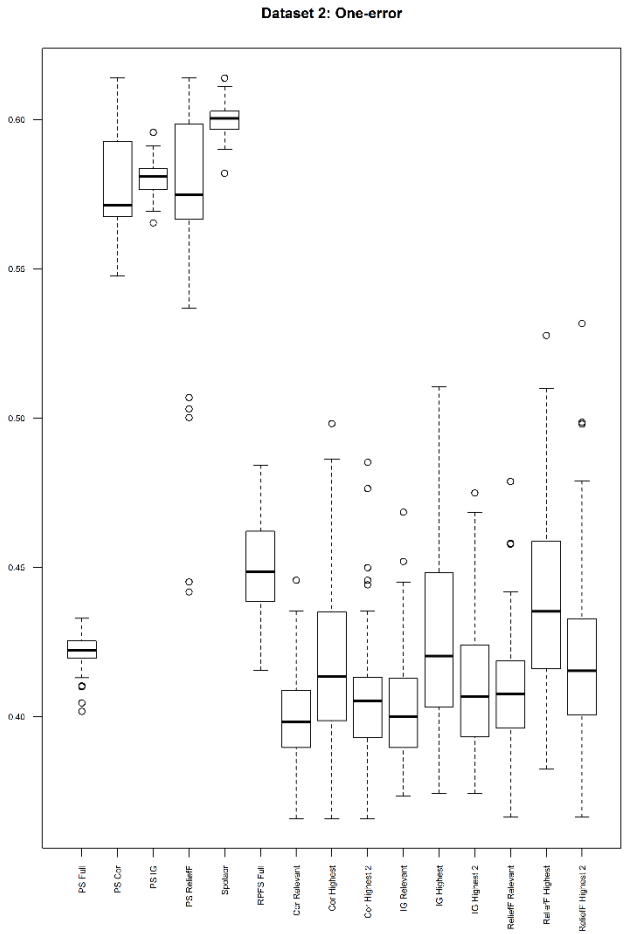


Figure J.6 Dataset 2: One-error.

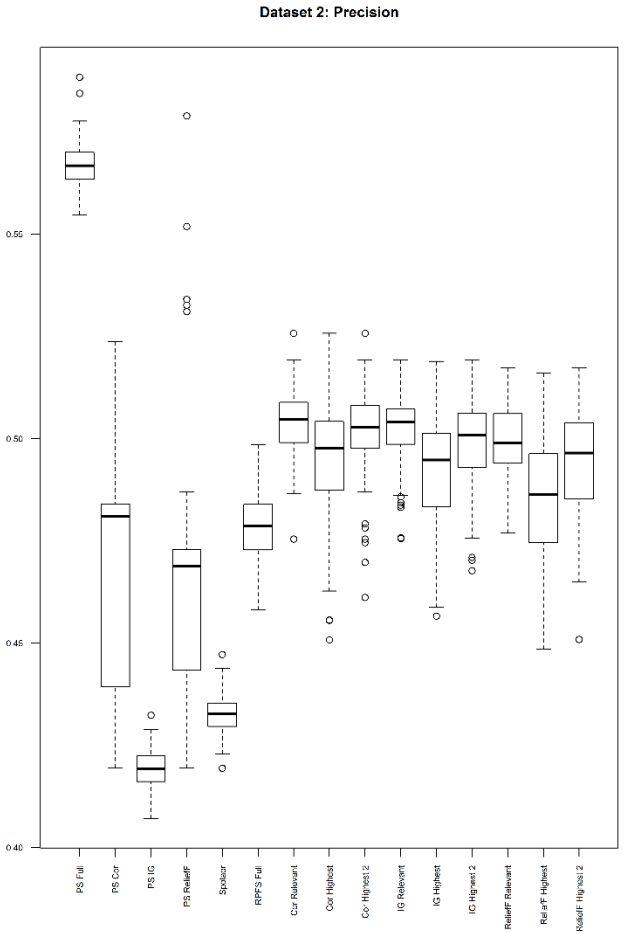


Figure J.7 Dataset 2: Precision.

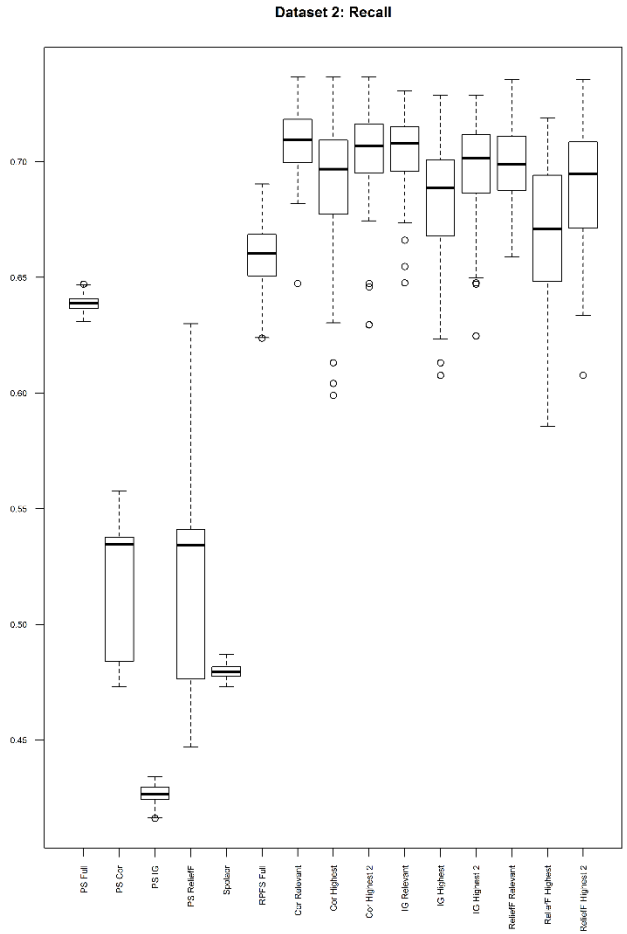


Figure J.8 Dataset 2: Recall.

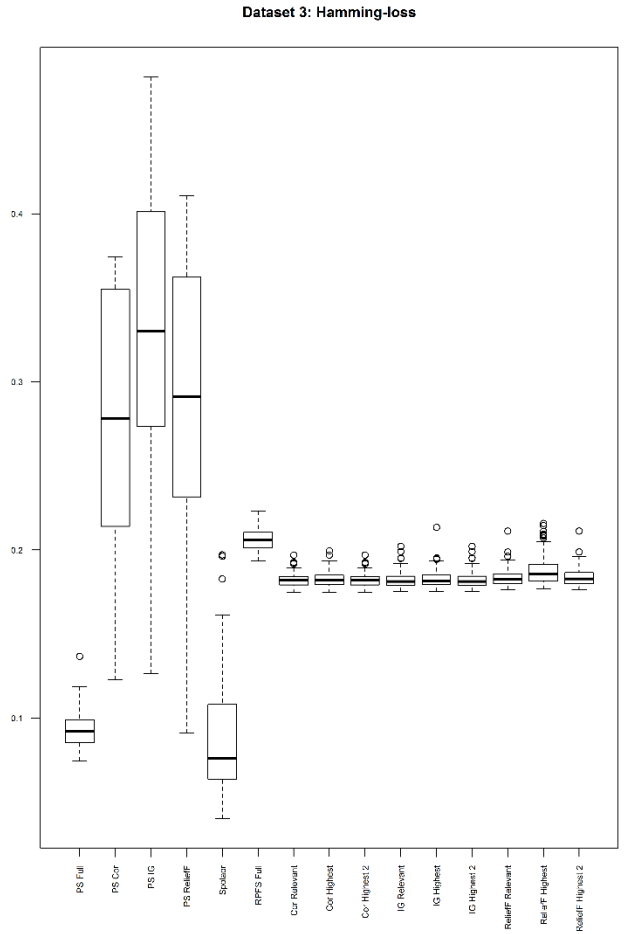


Figure J.9 Dataset 3: Hamming-loss.

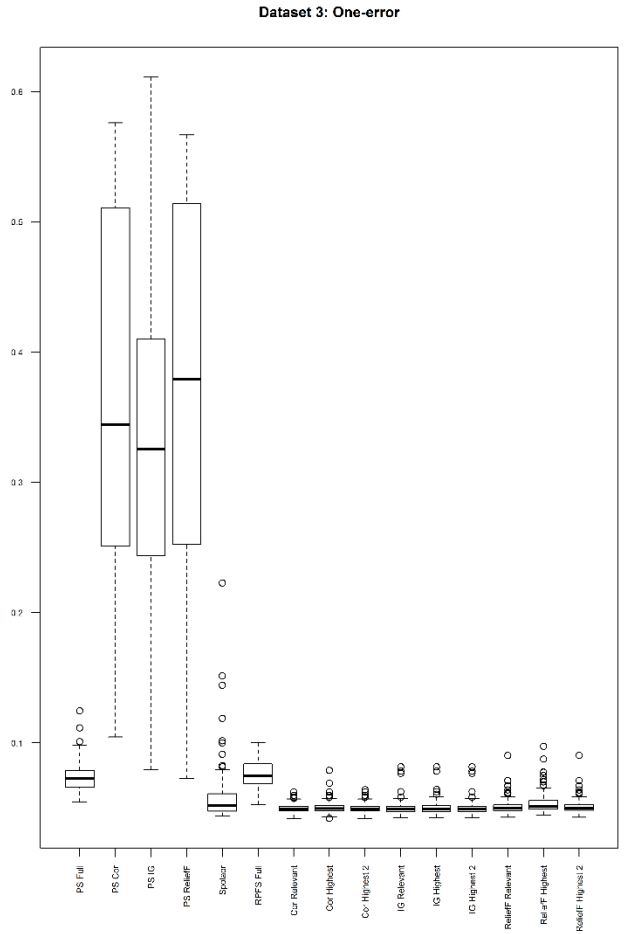


Figure J.10 Dataset 3: One-error.

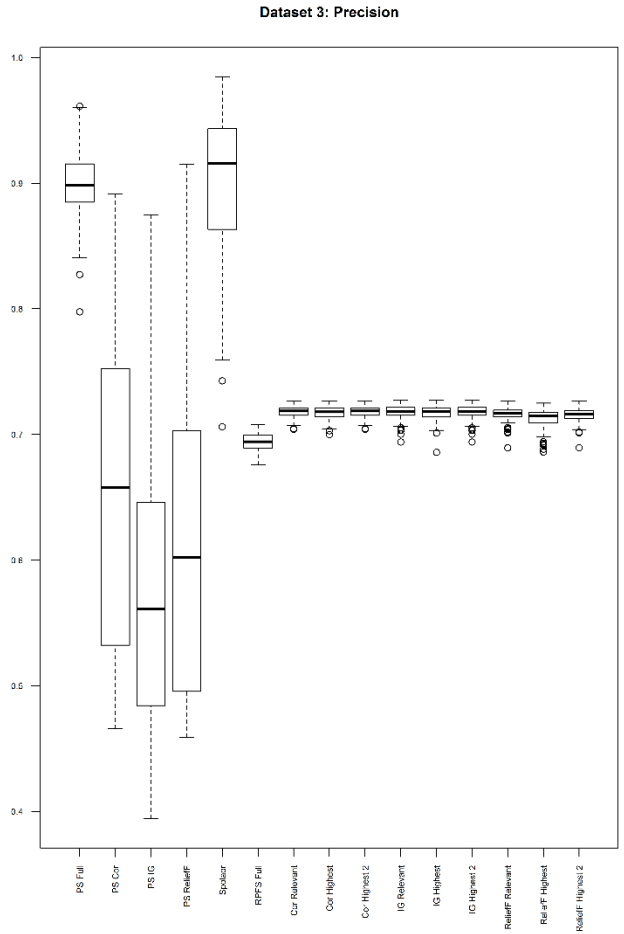


Figure J.11 Dataset 3: Precision.

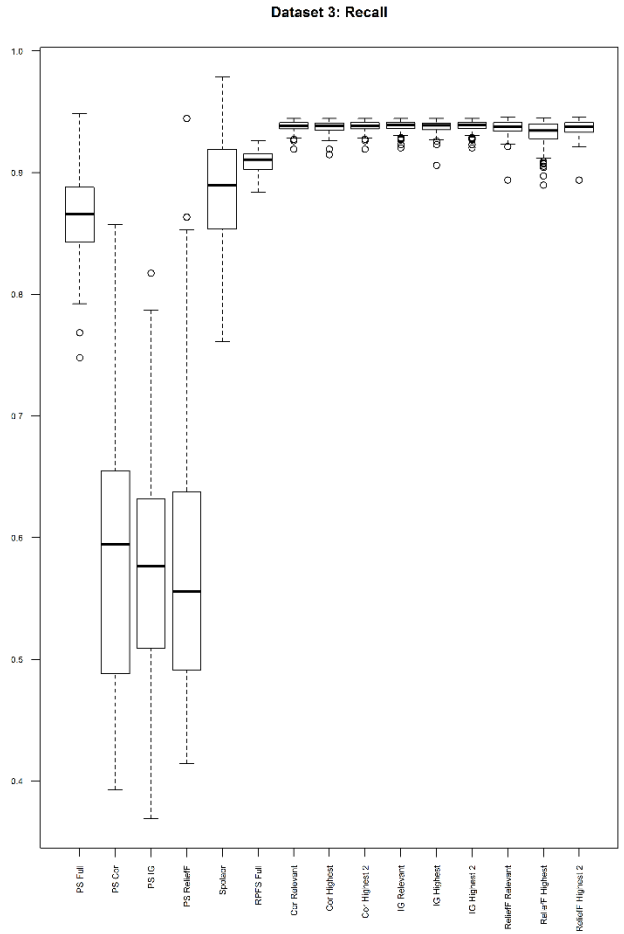


Figure J.12 Dataset 3: Recall.

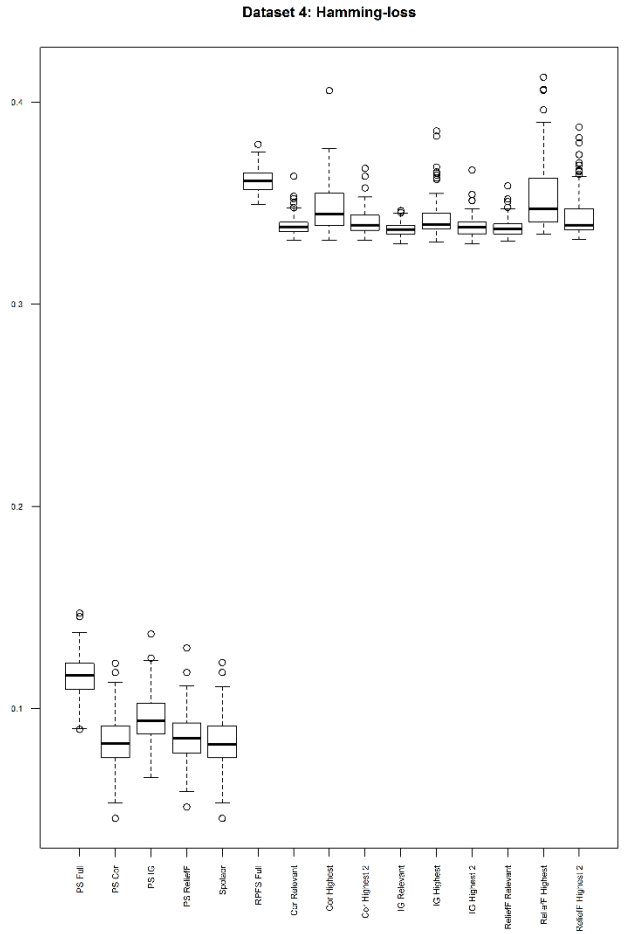


Figure J.13 Dataset 4: Hamming-loss.

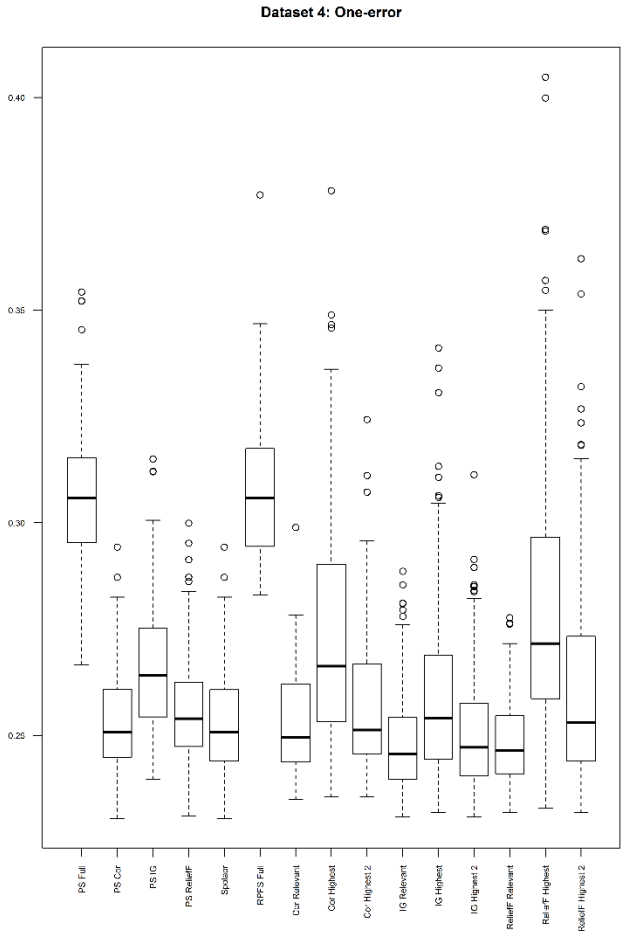


Figure J.14 Dataset 4: One-error.

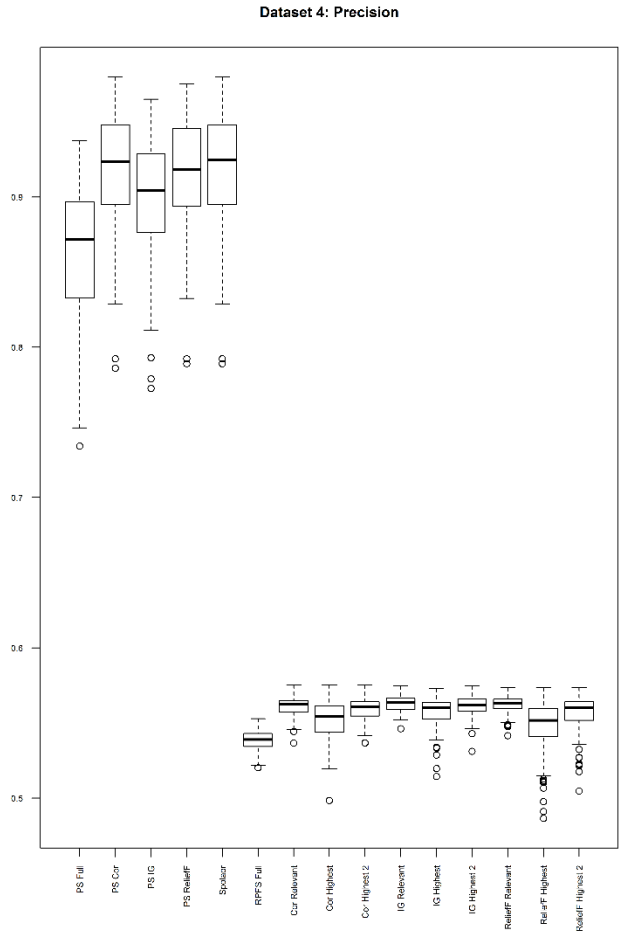


Figure J.15 Dataset 4: Precision.

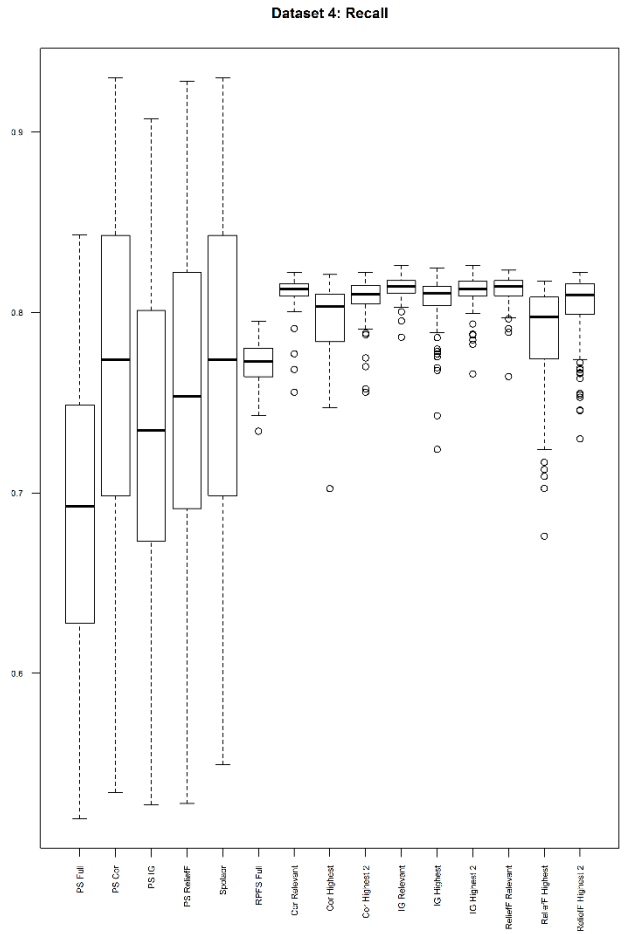


Figure J.16 Dataset 4: Recall.

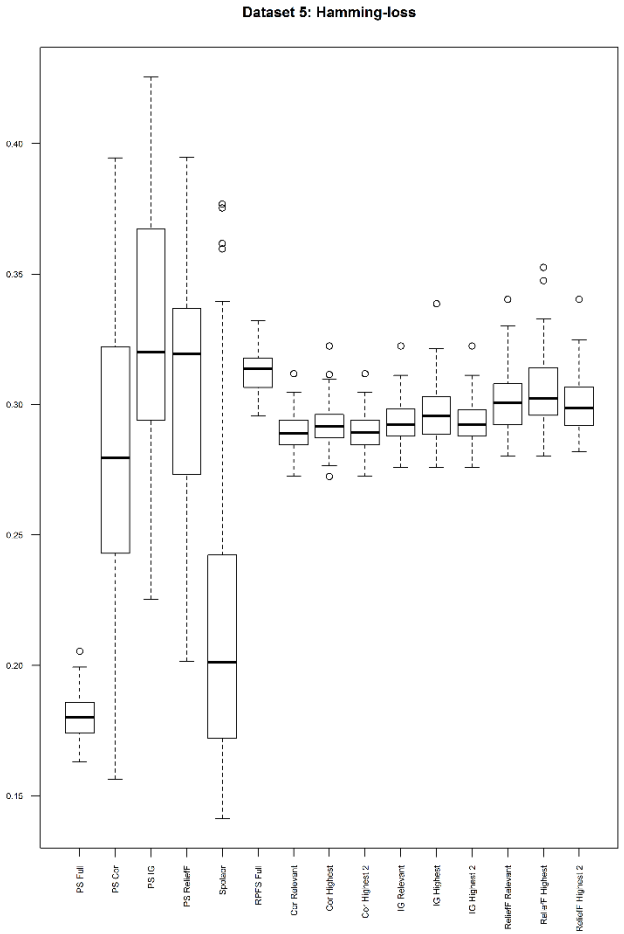


Figure J.17 Dataset 5: Hamming-loss.

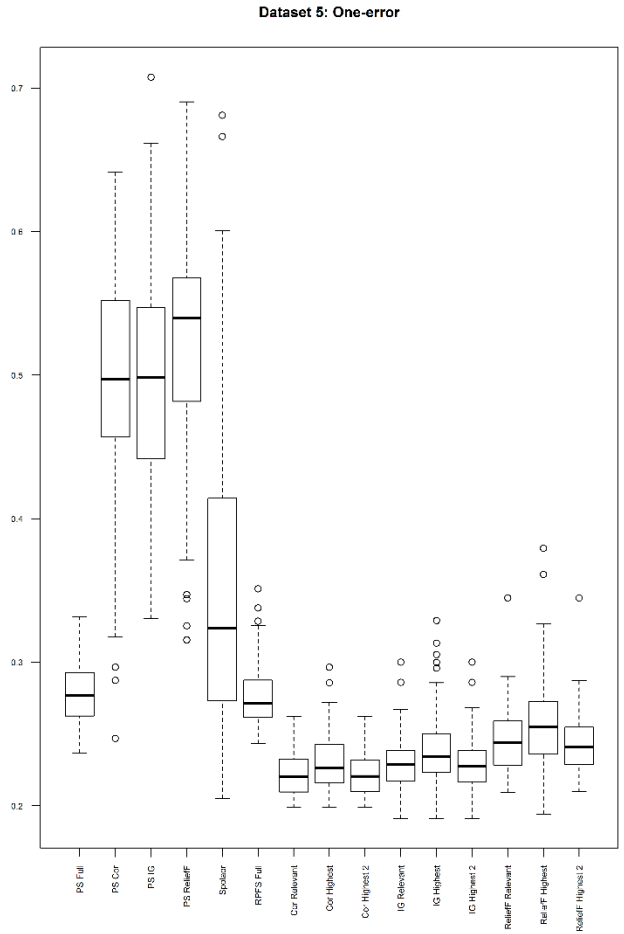


Figure J.18 Dataset 5: One-error.

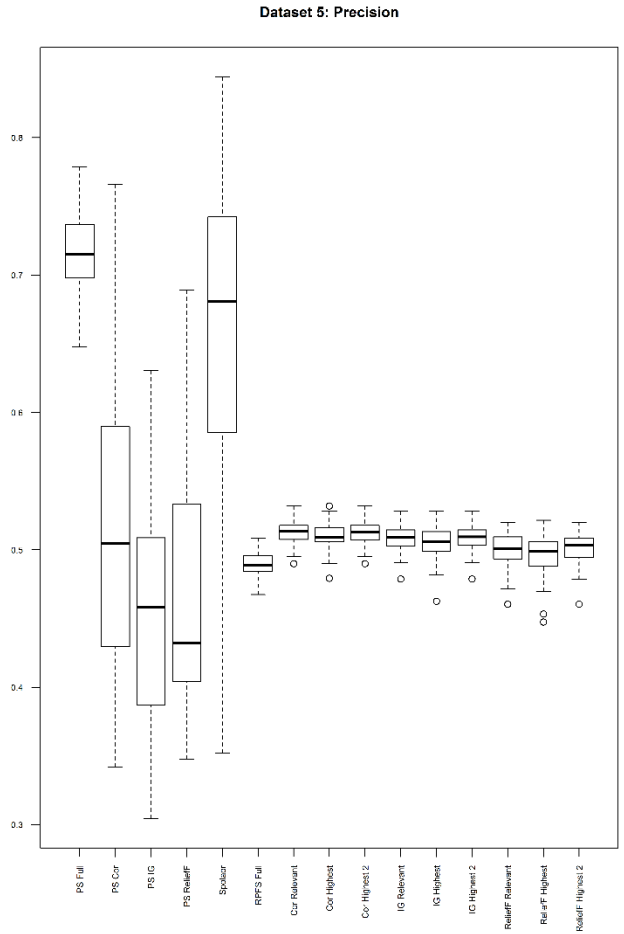


Figure J.19 Dataset 5: Precision.

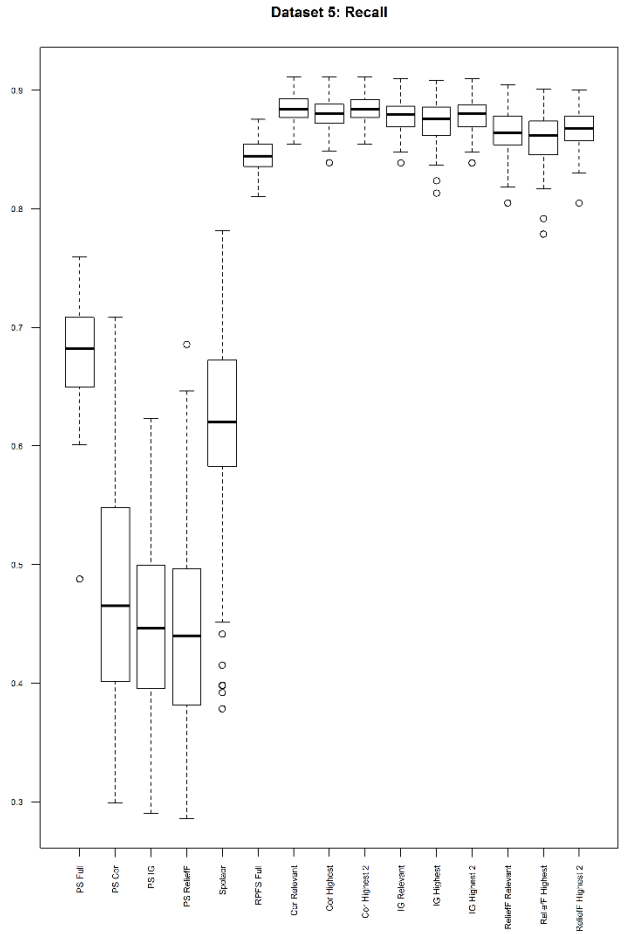


Figure J.20 Dataset 5: Recall.

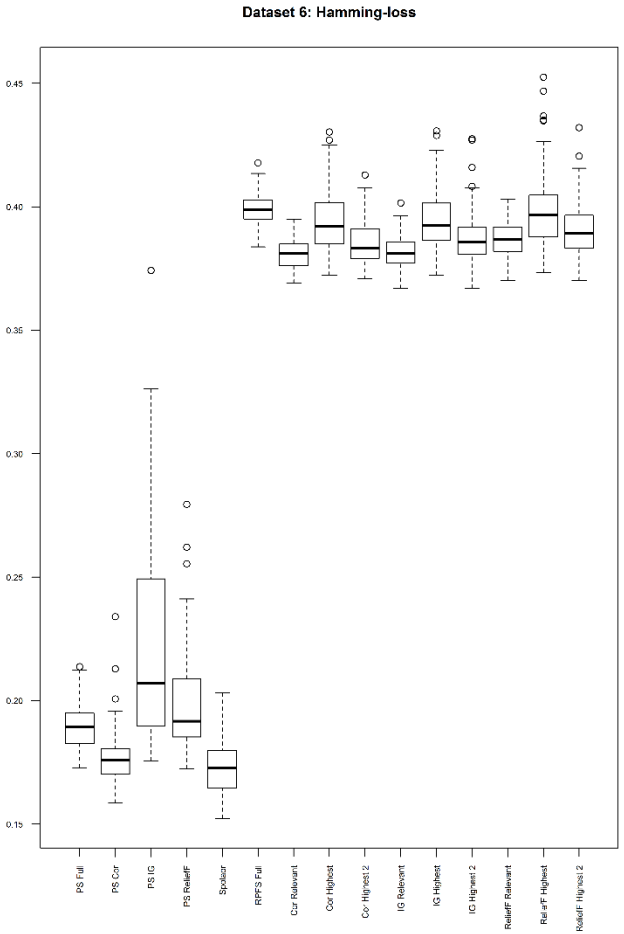


Figure J.21 Dataset 6: Hamming-loss.

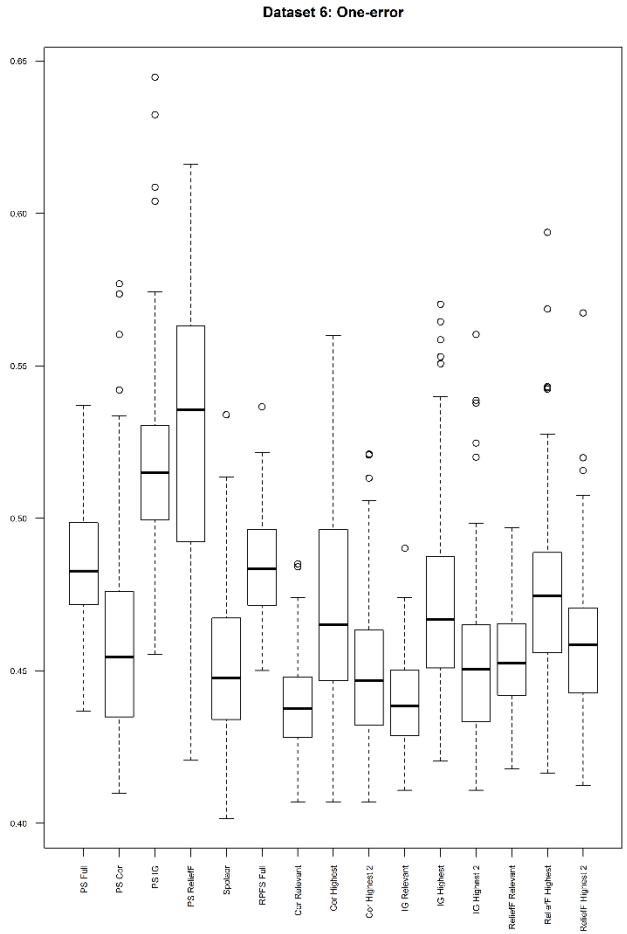


Figure J.22 Dataset 6: One-error.

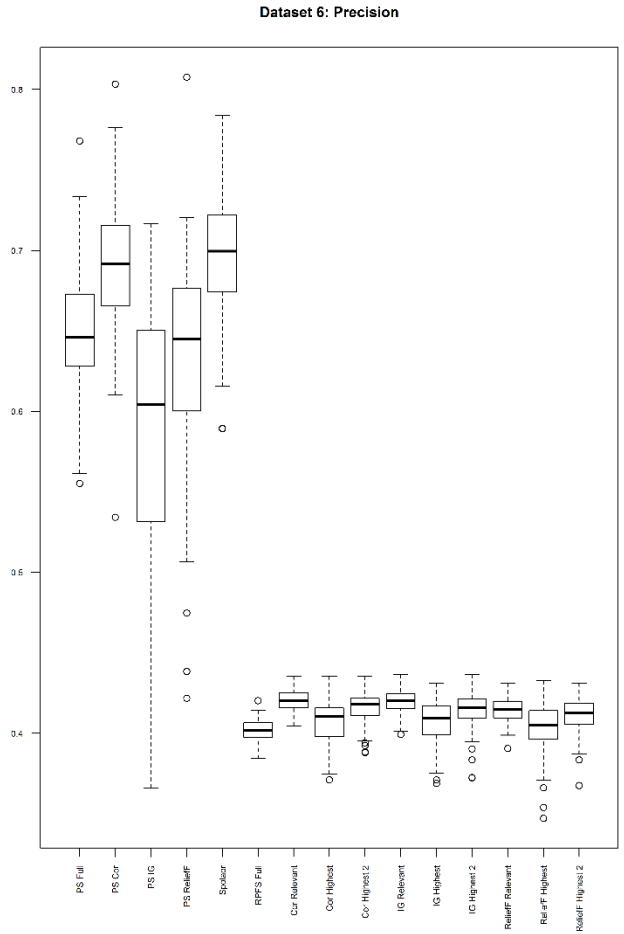


Figure J.23 Dataset 6: Precision.

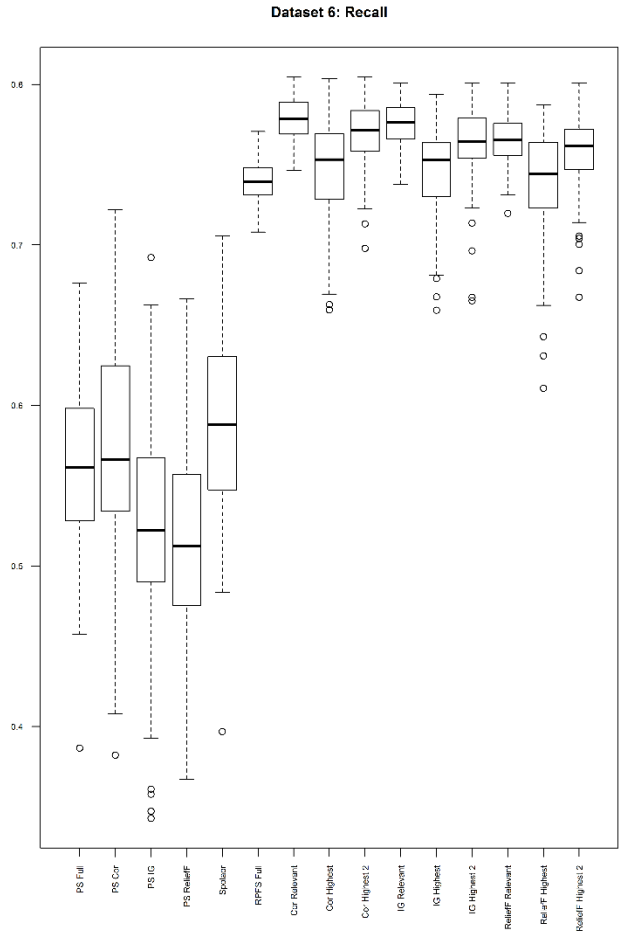


Figure J.24 Dataset 6: Recall.

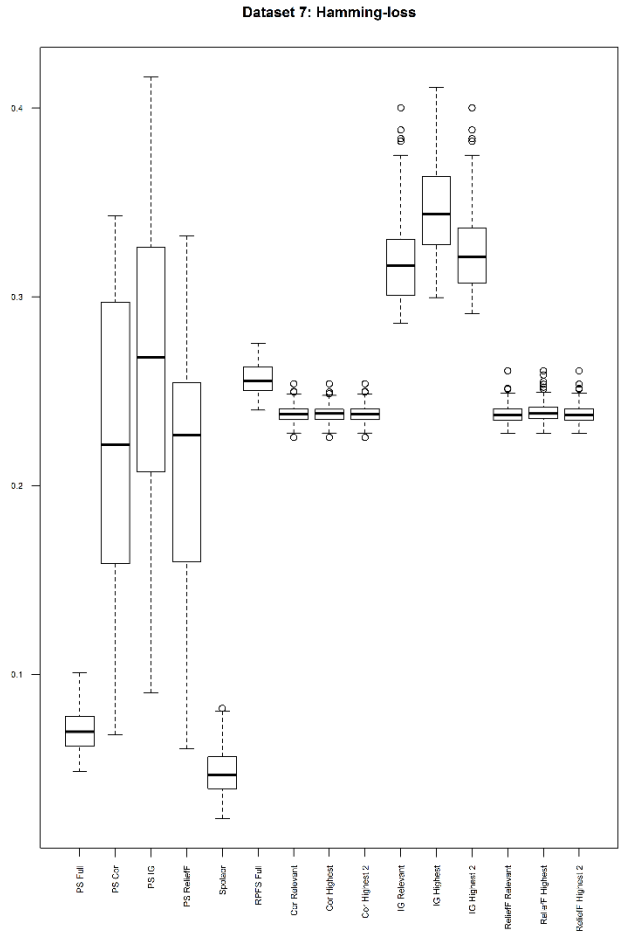


Figure J.25 Dataset 7: Hamming-loss.

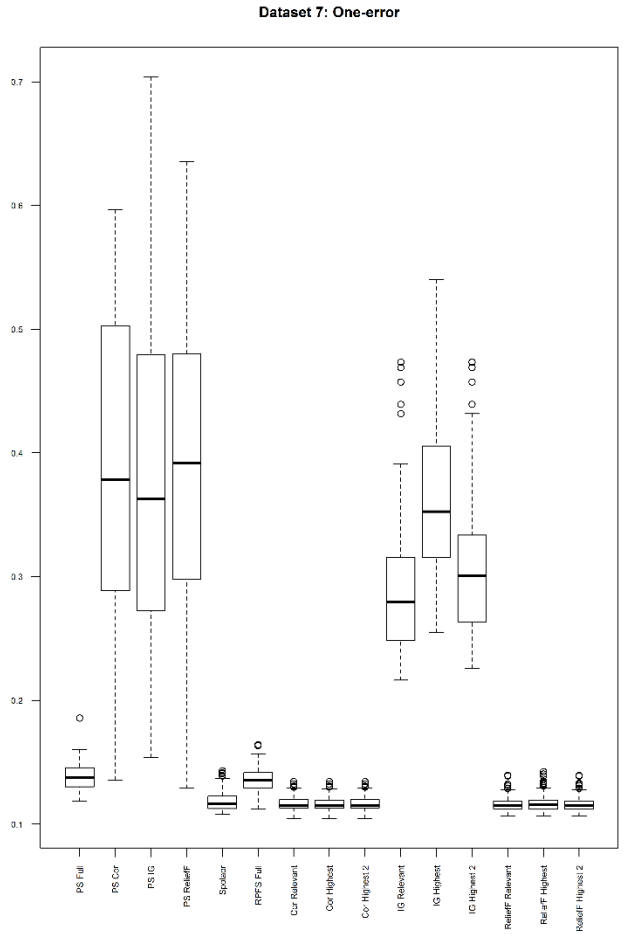


Figure J.26 Dataset 7: One-error.

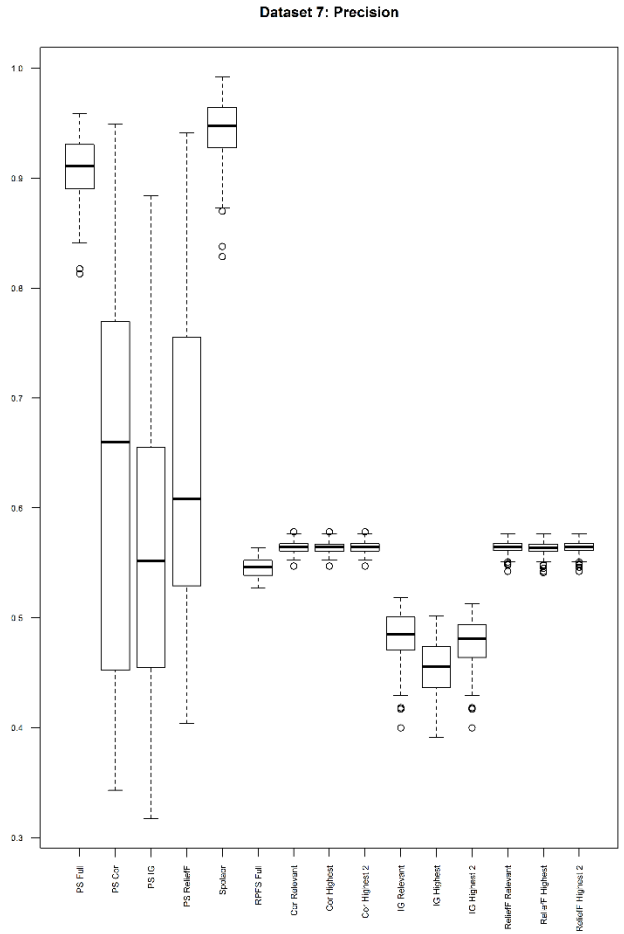


Figure J.27 Dataset 7: Precision.

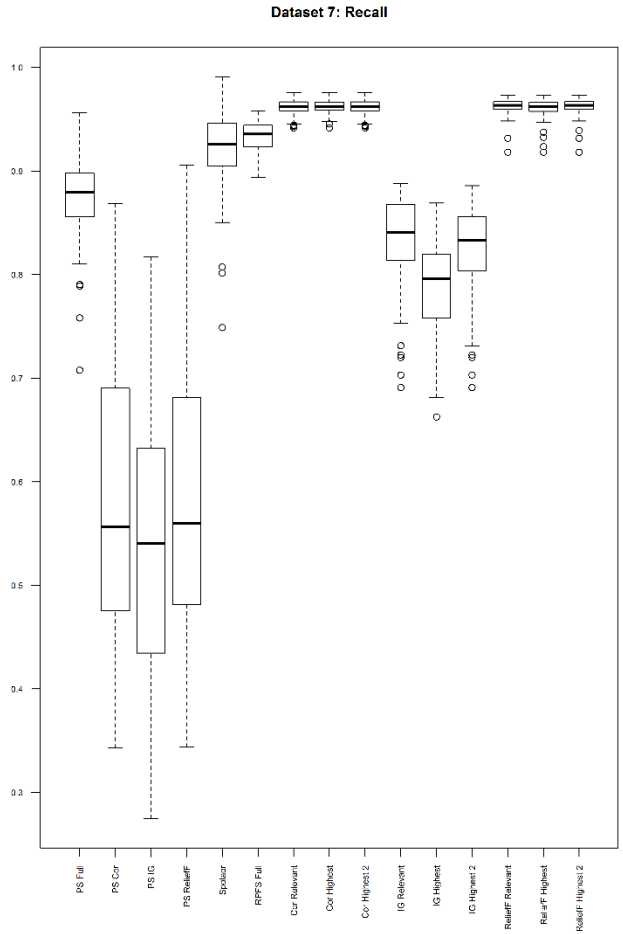


Figure J.28 Dataset 7: Recall.

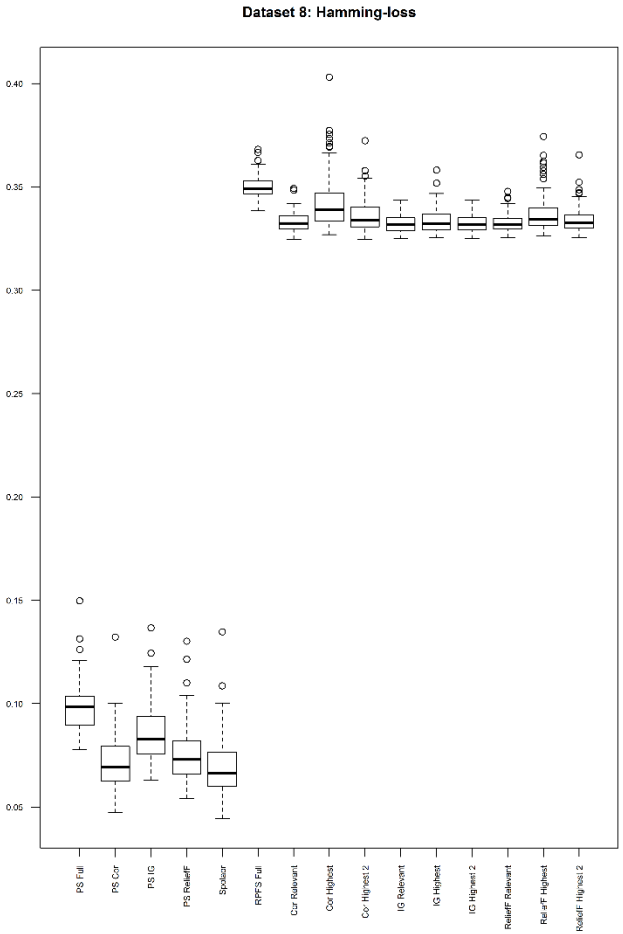


Figure J.29 Dataset 8: Hamming-loss.

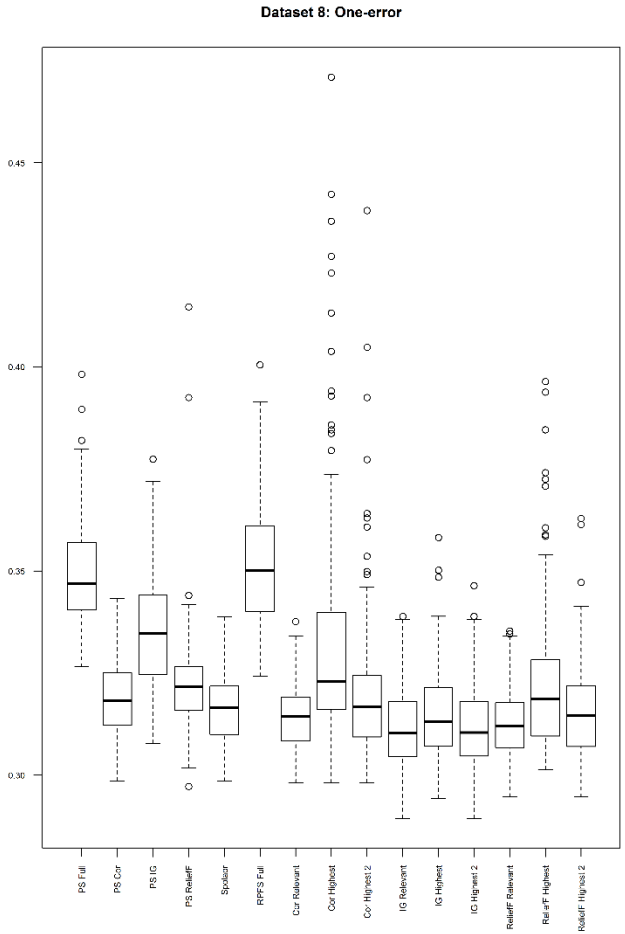


Figure J.30 Dataset 8: One-error.

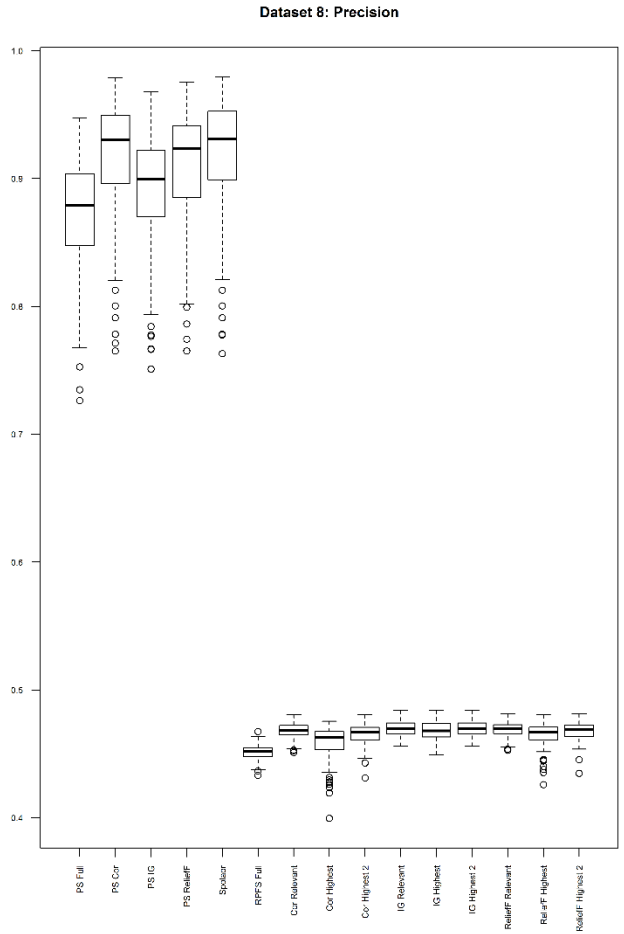


Figure J.31 Dataset 8: Precision.

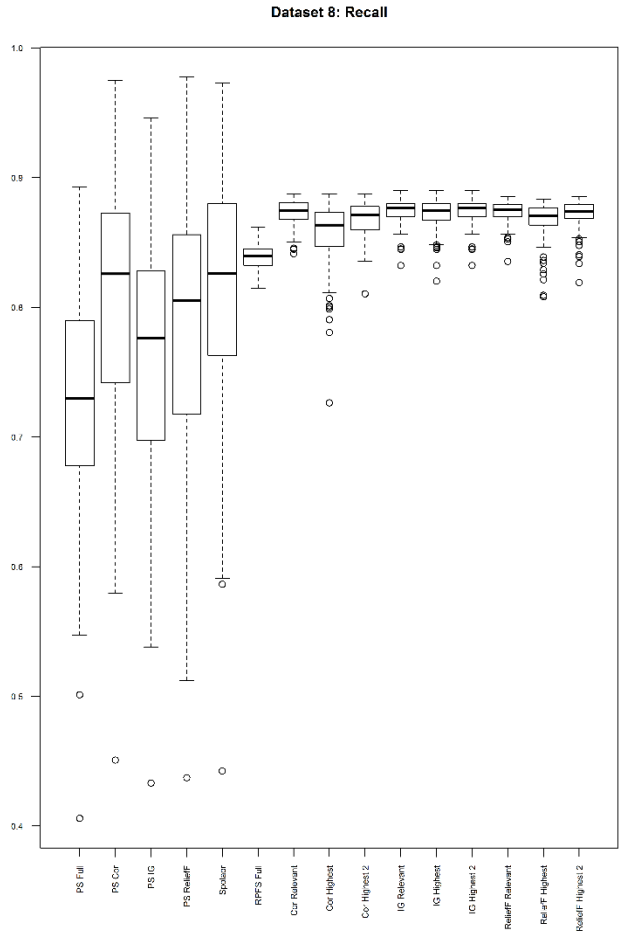


Figure J.32 Dataset 8: Recall.

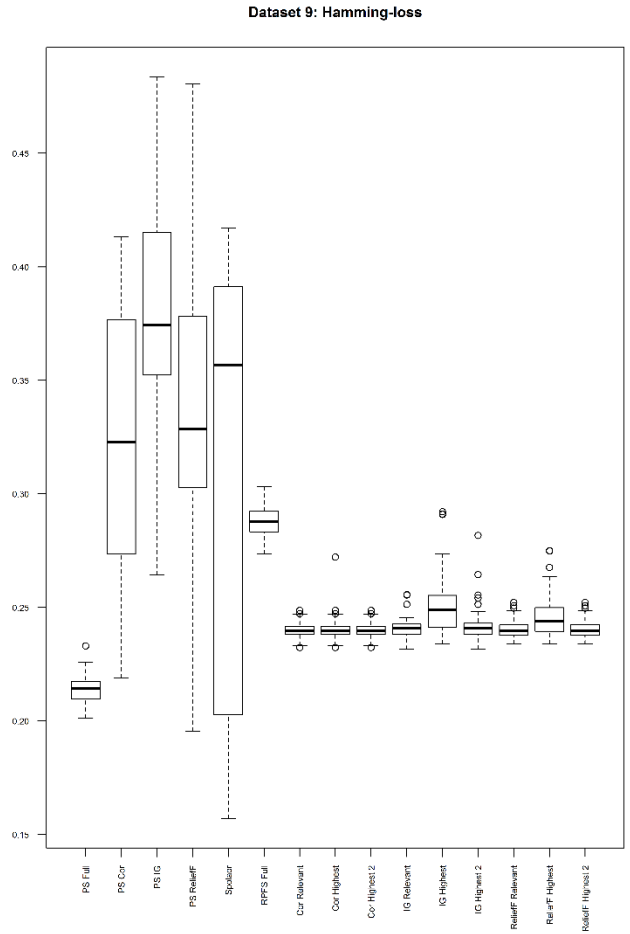


Figure J.33 Dataset 9: Hamming-loss.

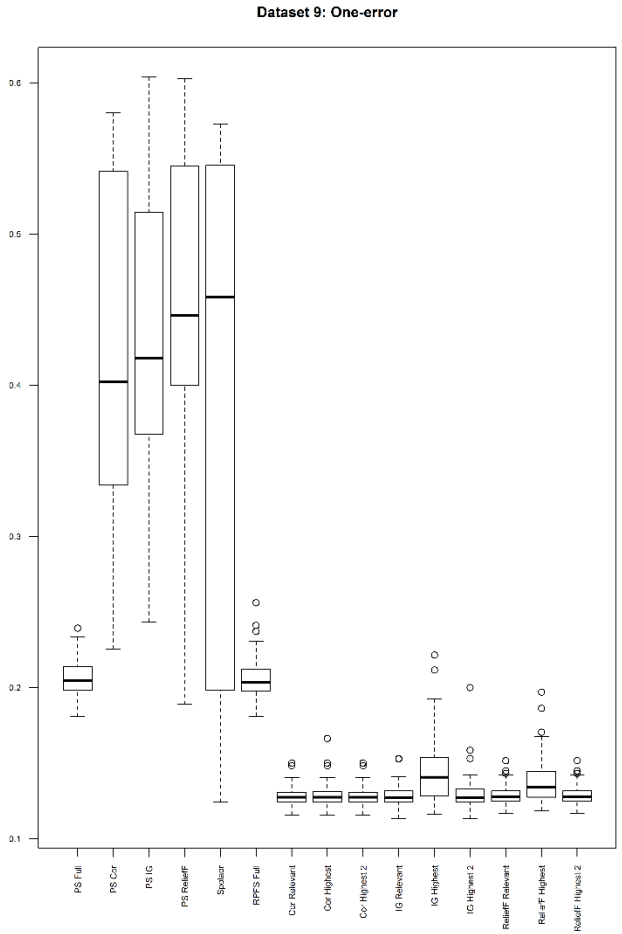


Figure J.34 Dataset 9: One-error.

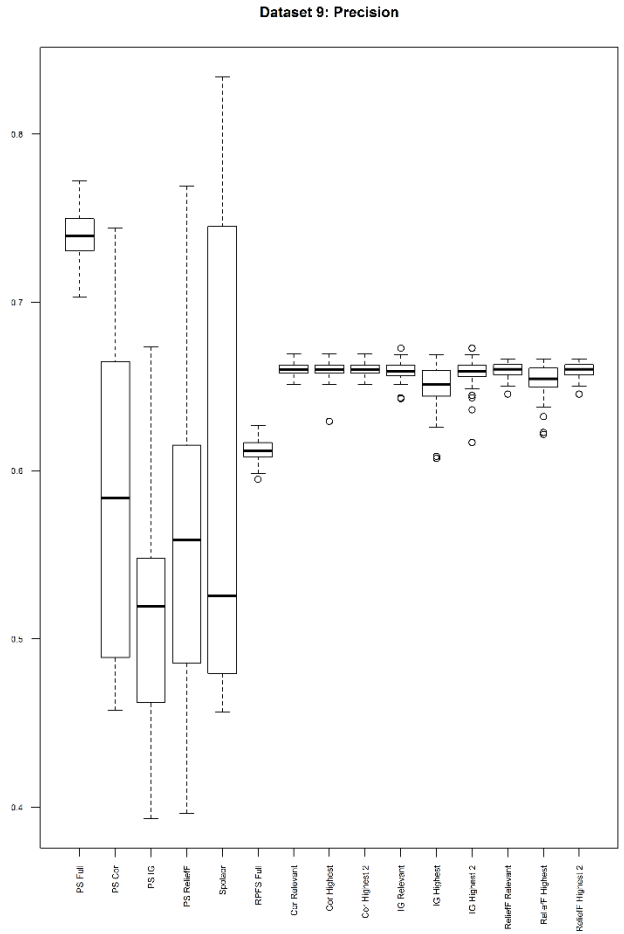


Figure J.35 Dataset 9: Precision.

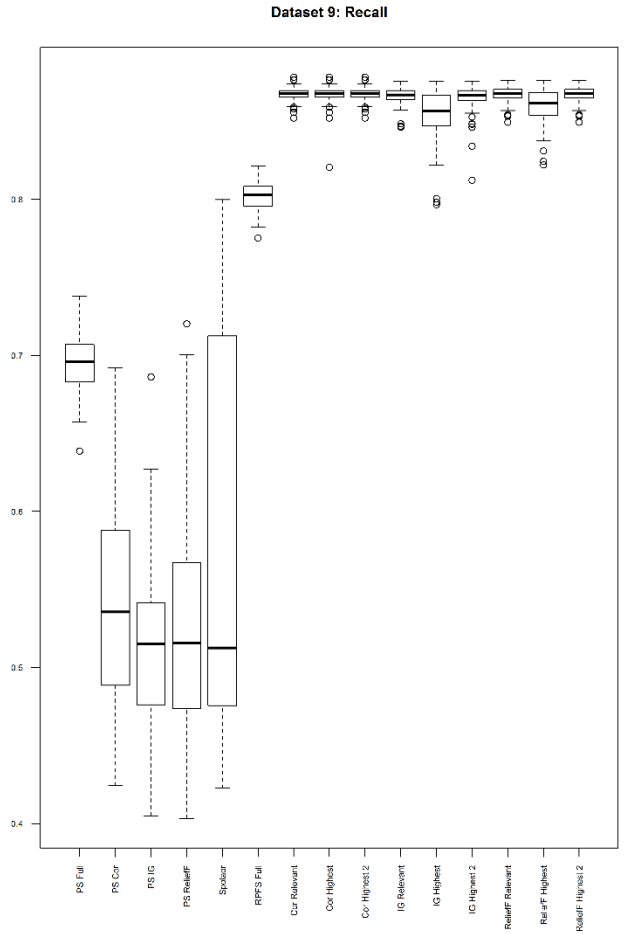


Figure J.36 Dataset 9: Recall.

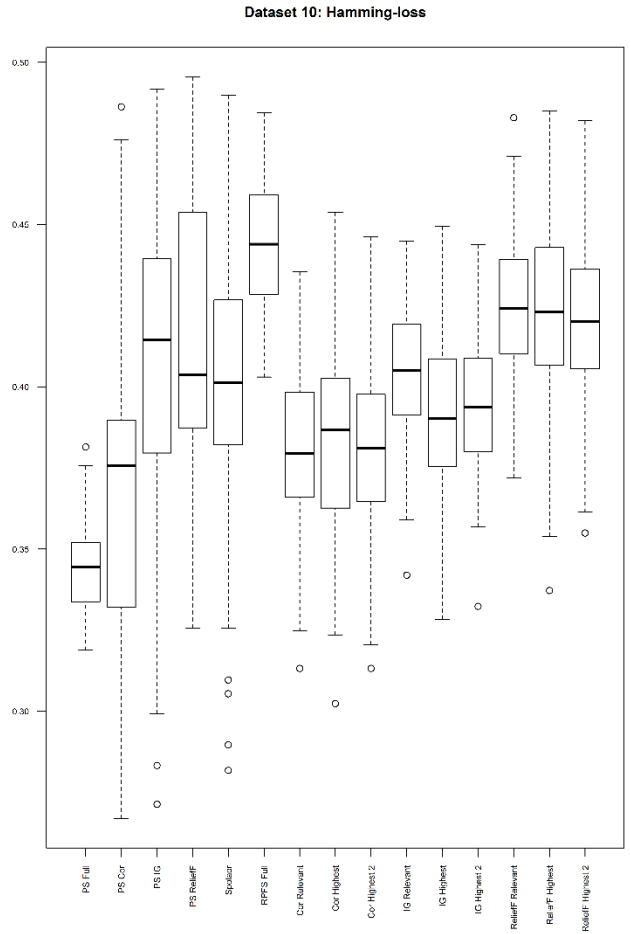


Figure J.37 Dataset 10: Hamming-loss.

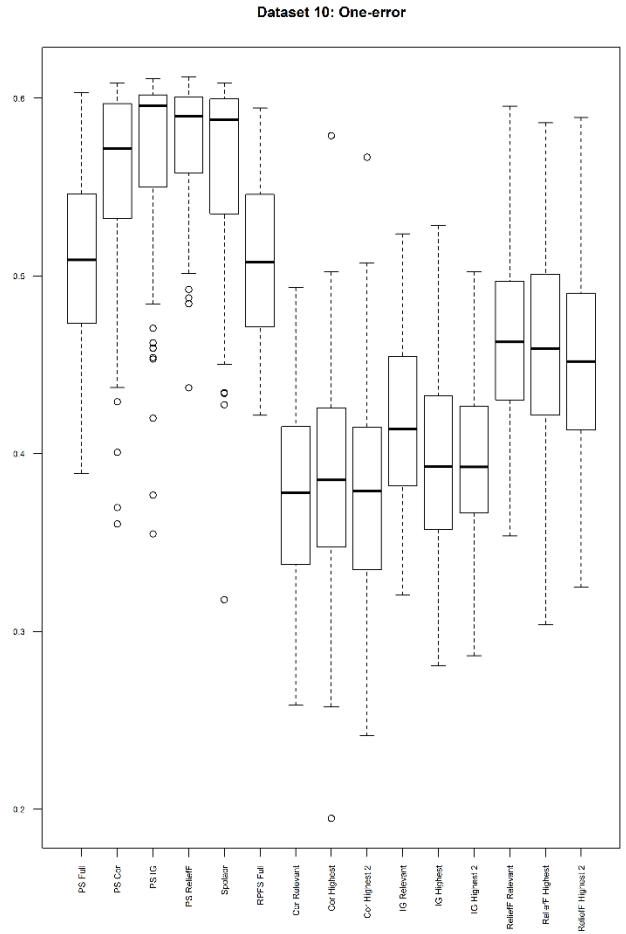


Figure J.38 Dataset 10: One-error.

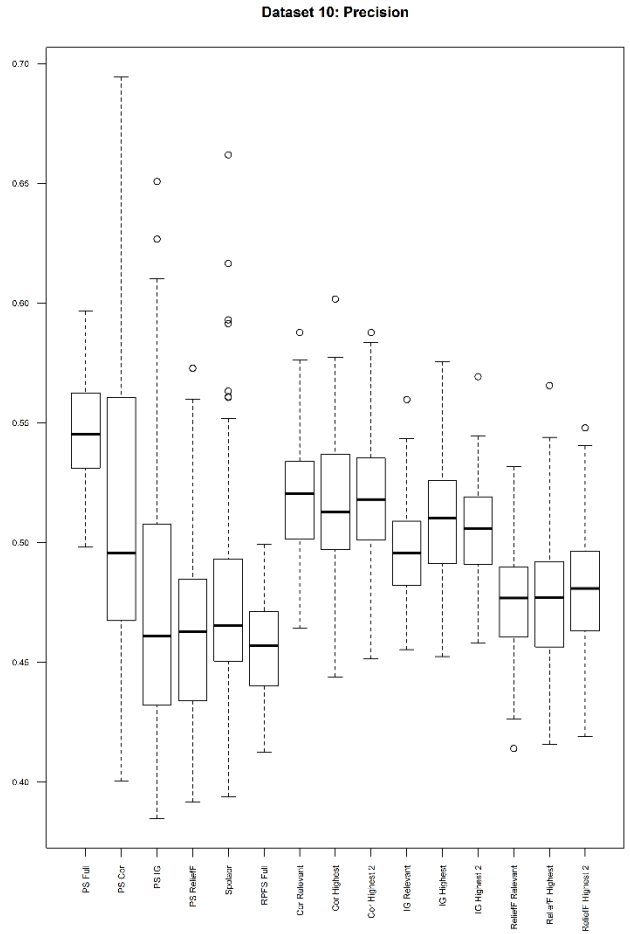


Figure J.39 Dataset 10: Precision.

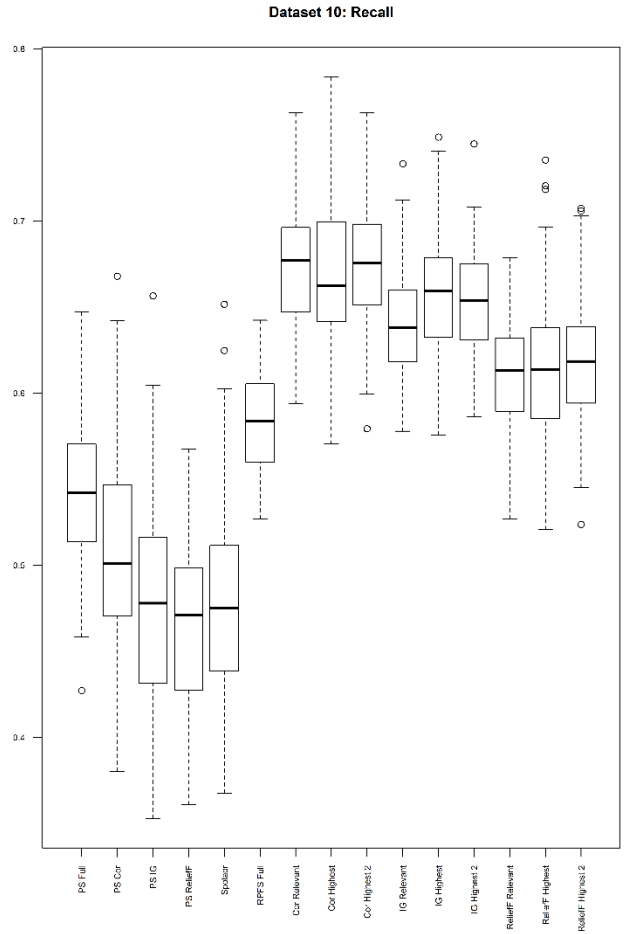


Figure J.40 Dataset 10: Recall.

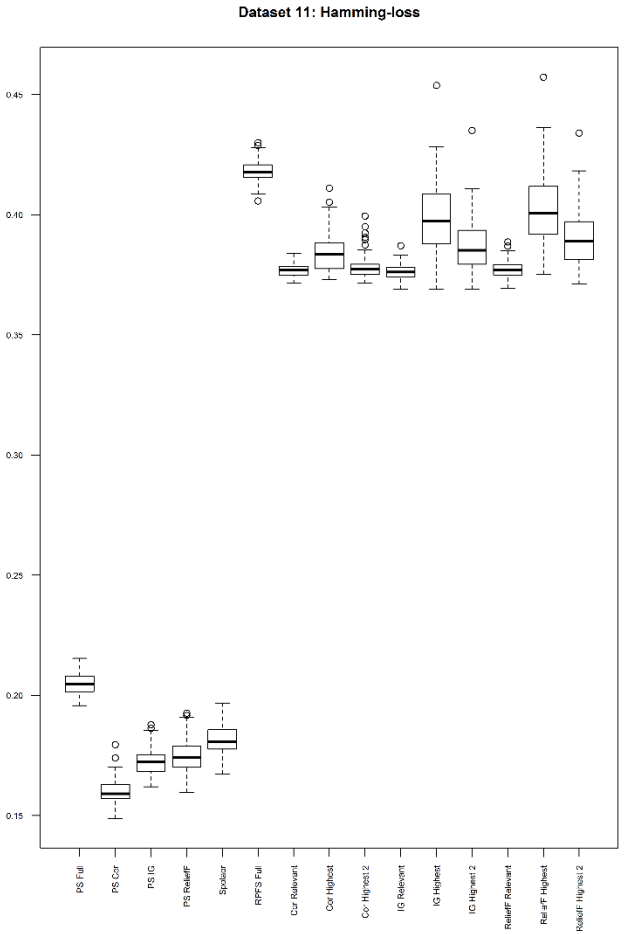


Figure J.41 Dataset 11: Hamming-loss.

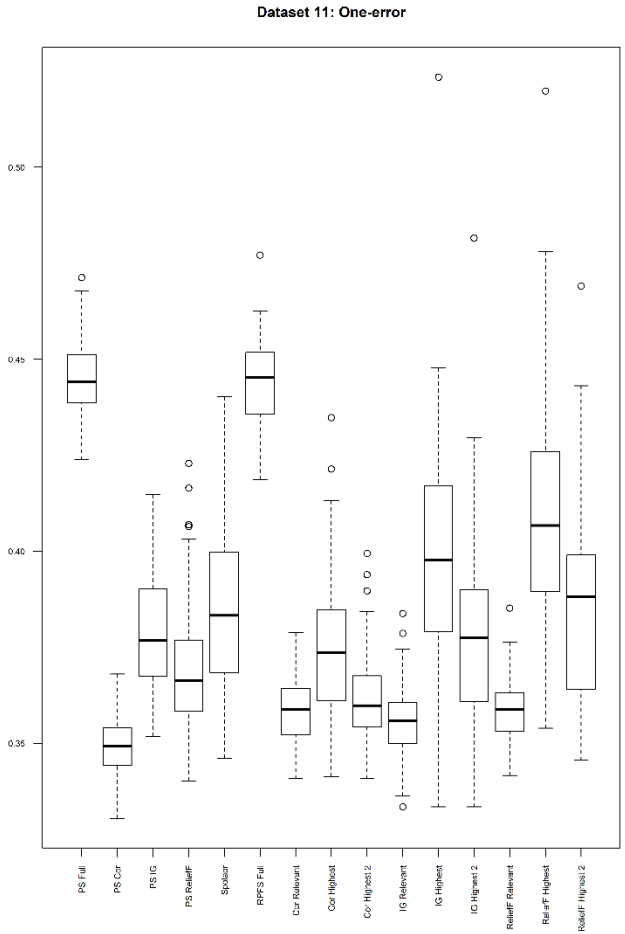


Figure J.42 Dataset 11: One-error.

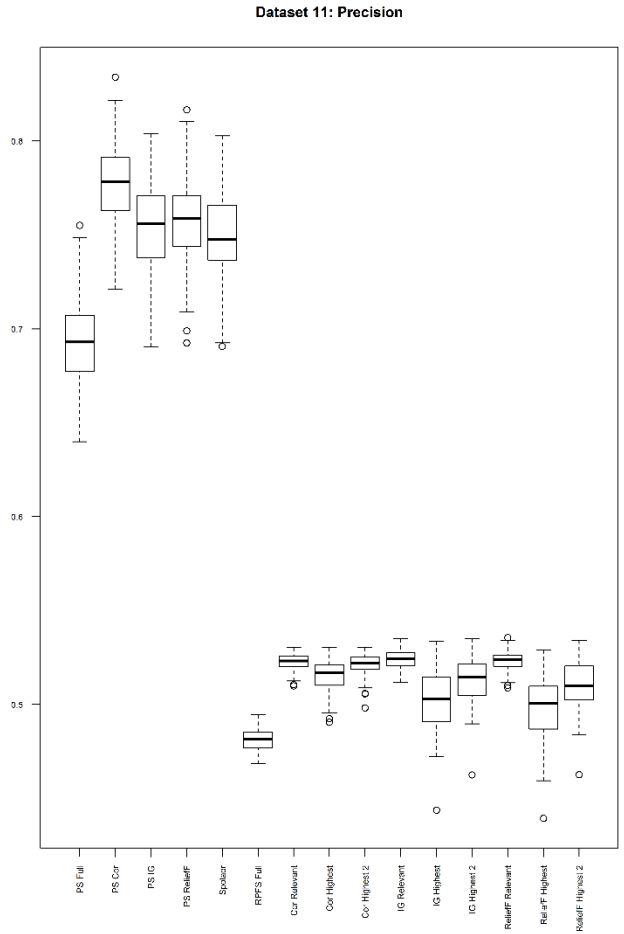


Figure J.43 Dataset 11: Precision.

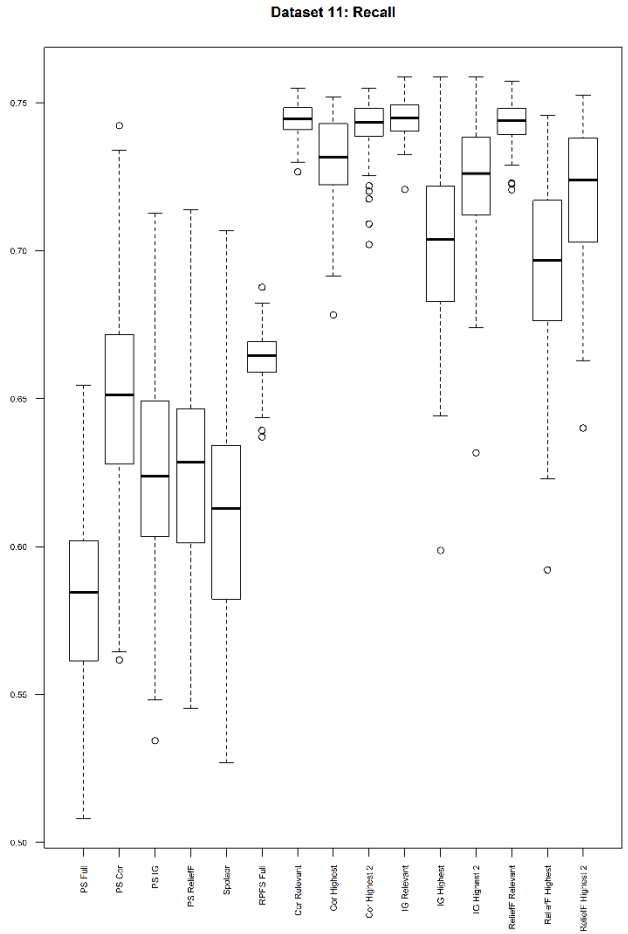


Figure J.44 Dataset 11: Recall.

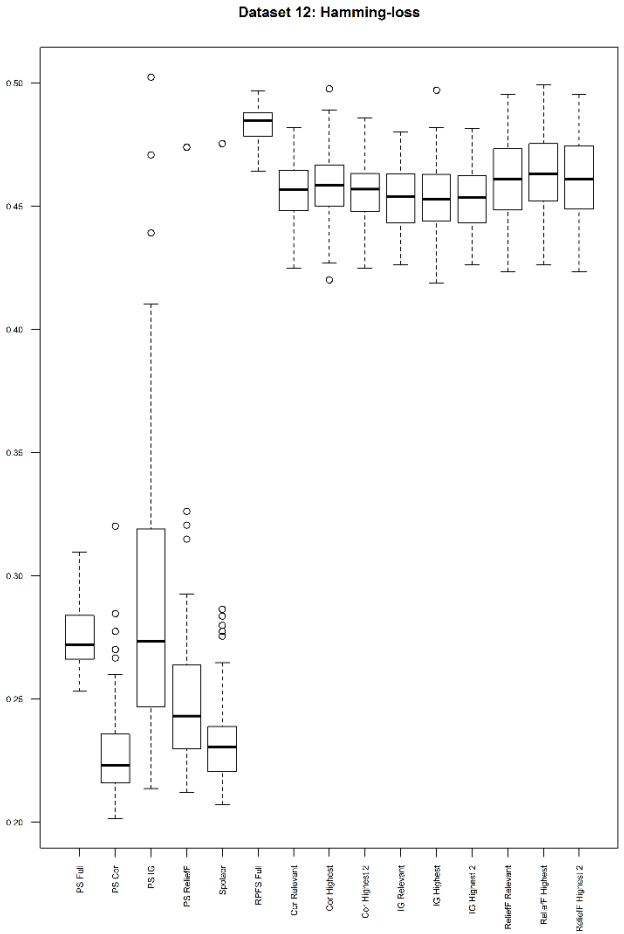


Figure J.45 Dataset 12: Hamming-loss.

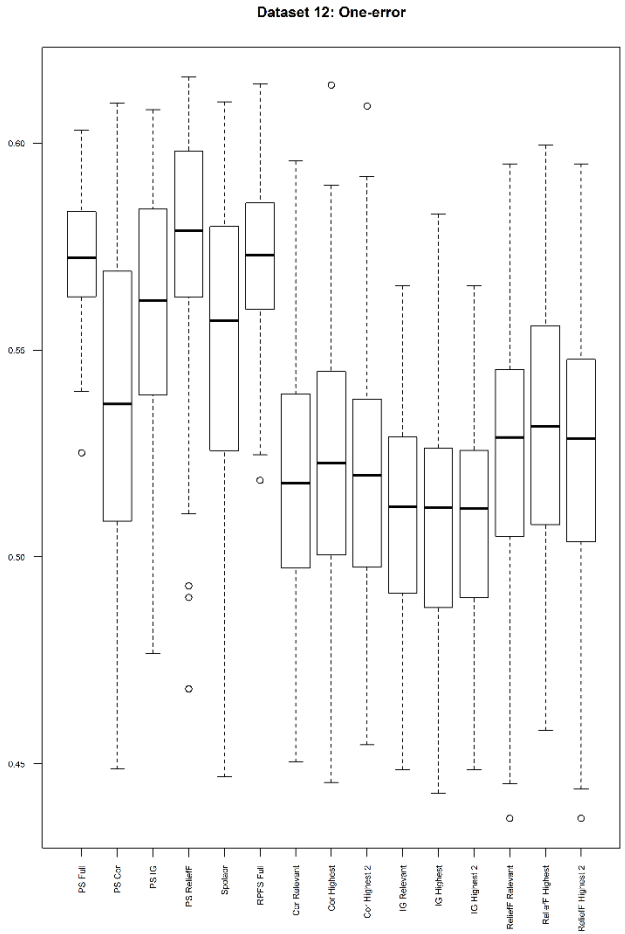


Figure J.46 Dataset 12: One-error.

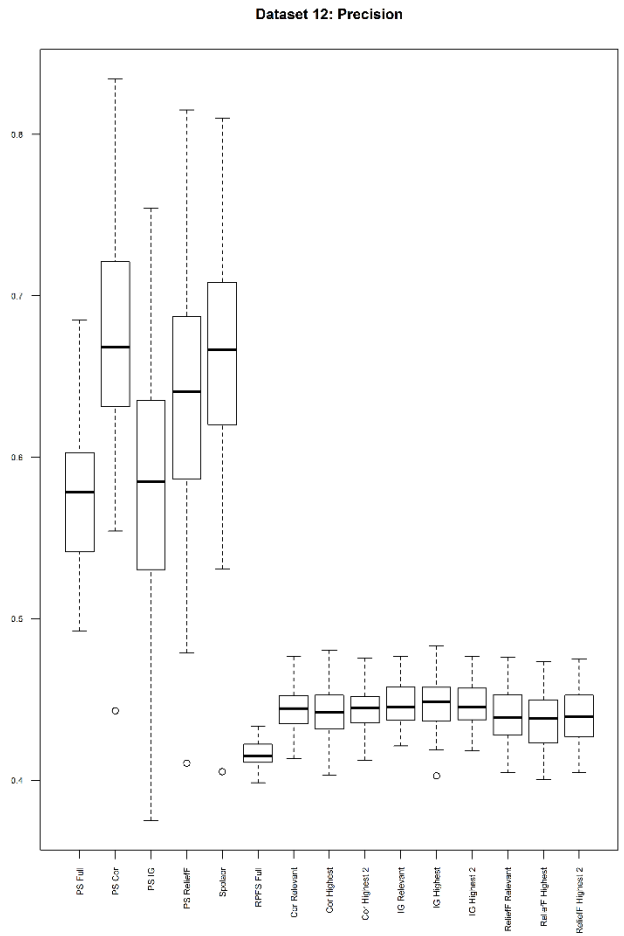


Figure J.47 Dataset 12: Precision.

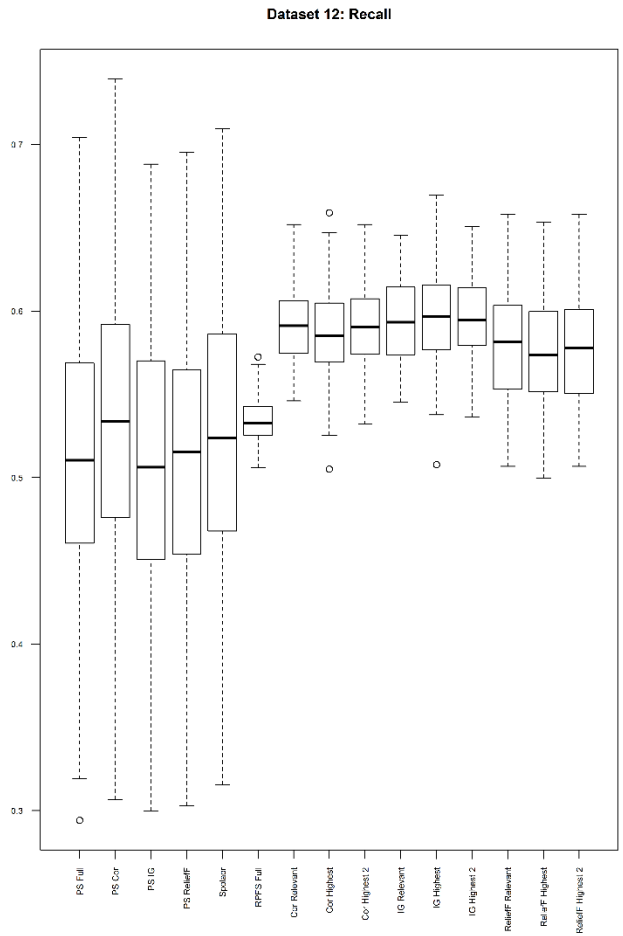


Figure J.48 Dataset 12: Recall.

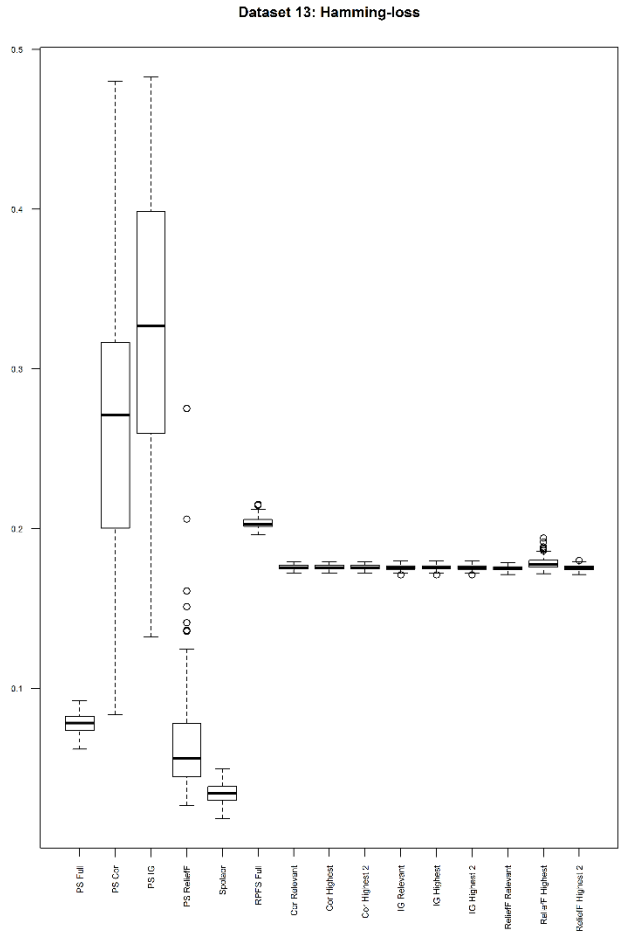


Figure J.49 Dataset 13: Hamming-loss.

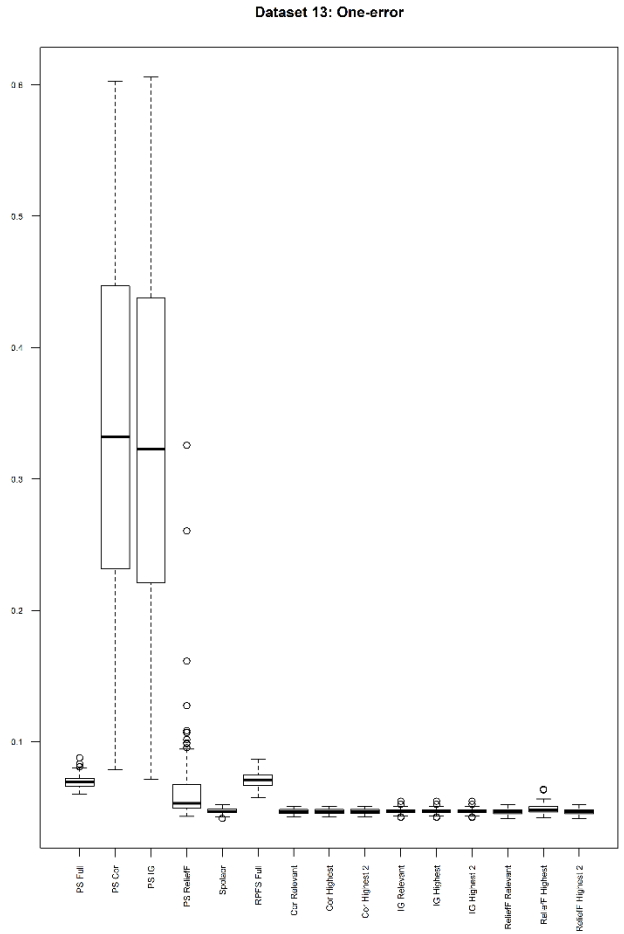


Figure J.50 Dataset 13: One-error.

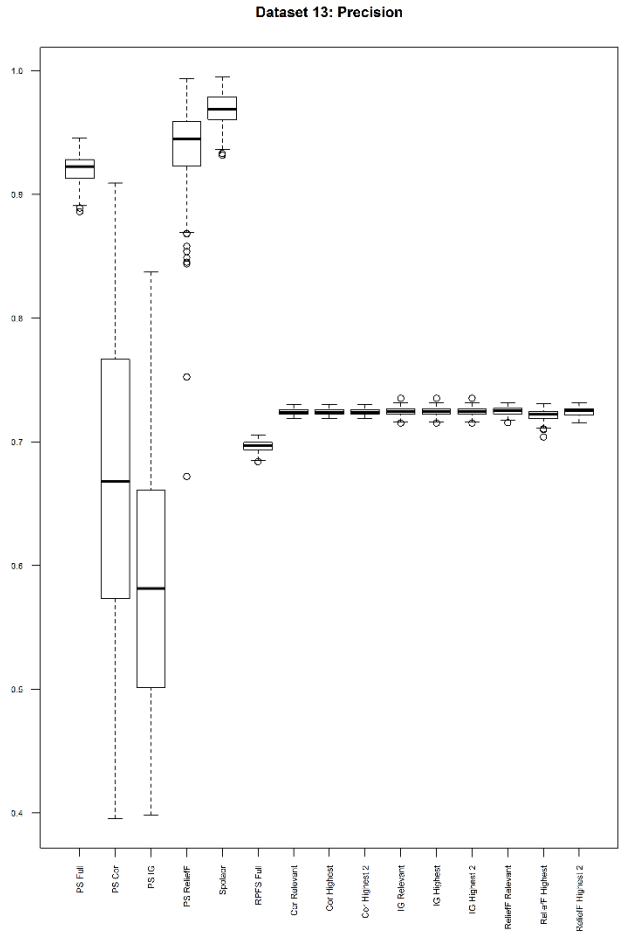


Figure J.51 Dataset 13: Precision.

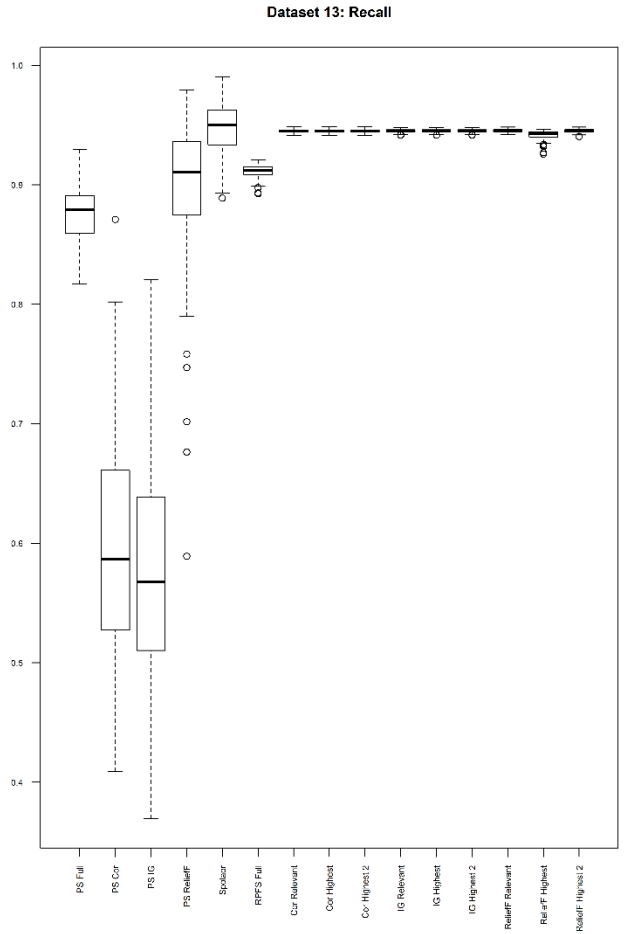


Figure J.52 Dataset 13: Recall.

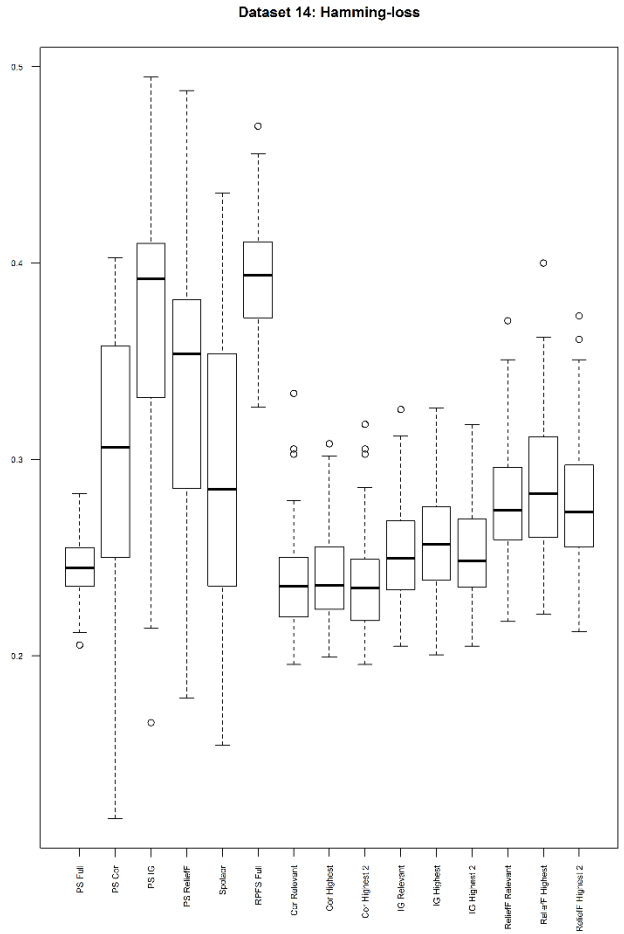


Figure J.53 Dataset 14: Hamming-loss.

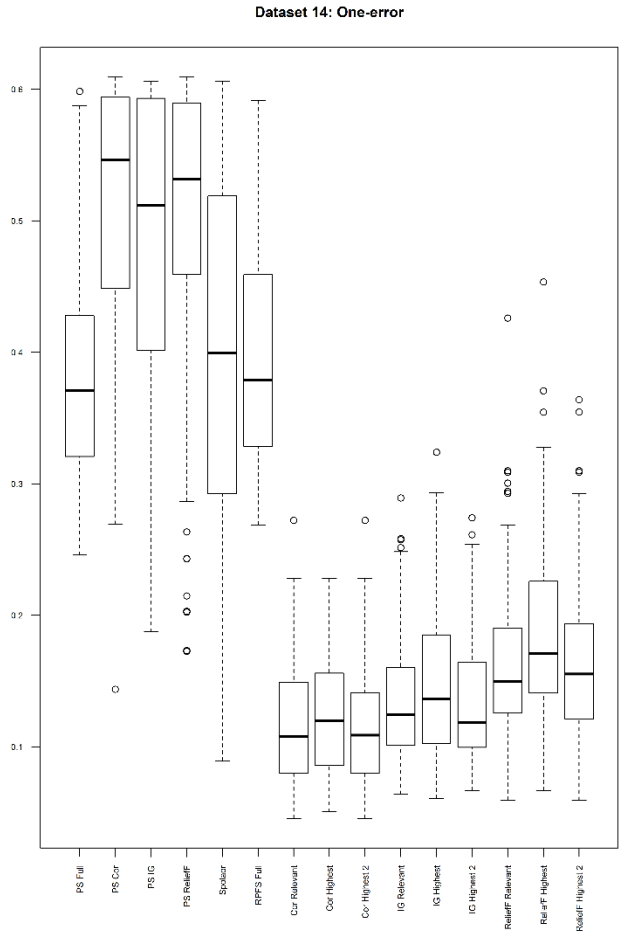


Figure J.54 Dataset 14: One-error.

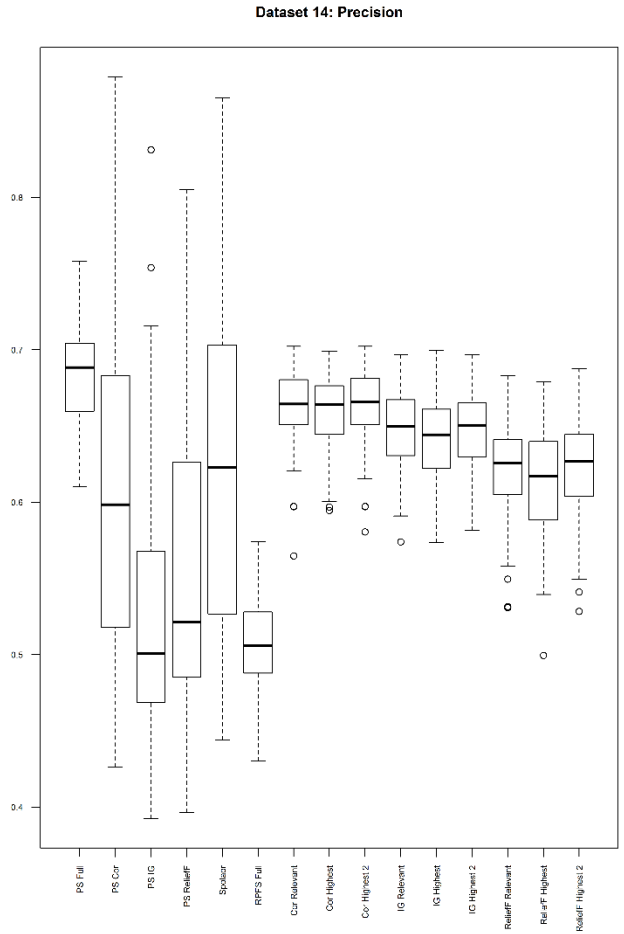


Figure J.55 Dataset 14: Precision.

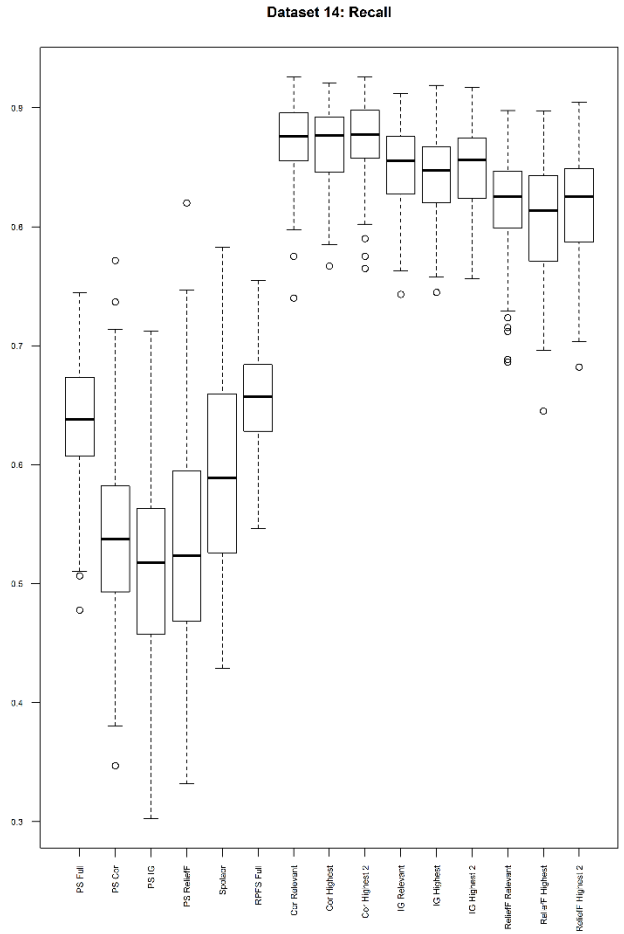


Figure J.56 Dataset 14: Recall.

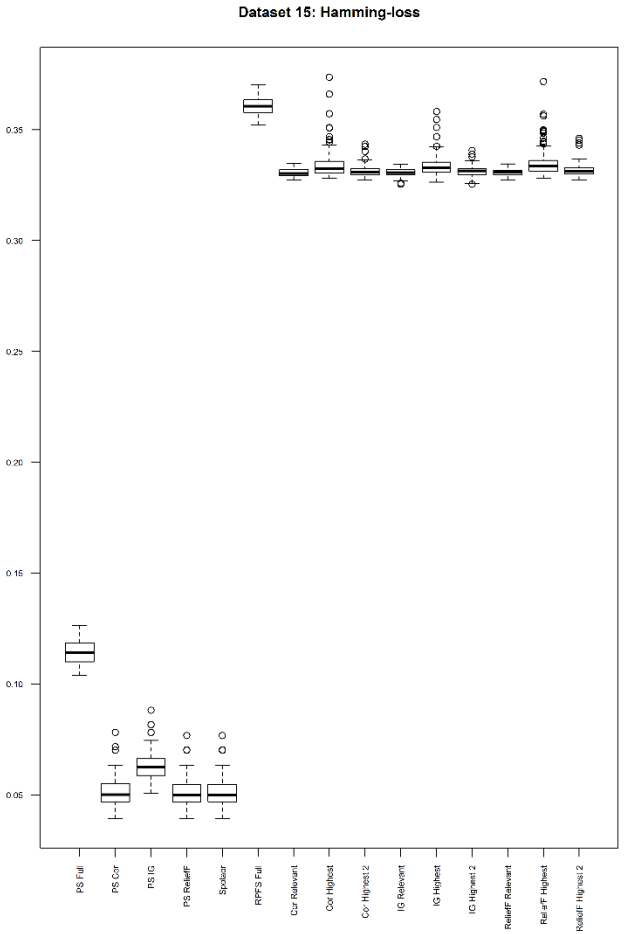


Figure J.57 Dataset 15: Hamming-loss.

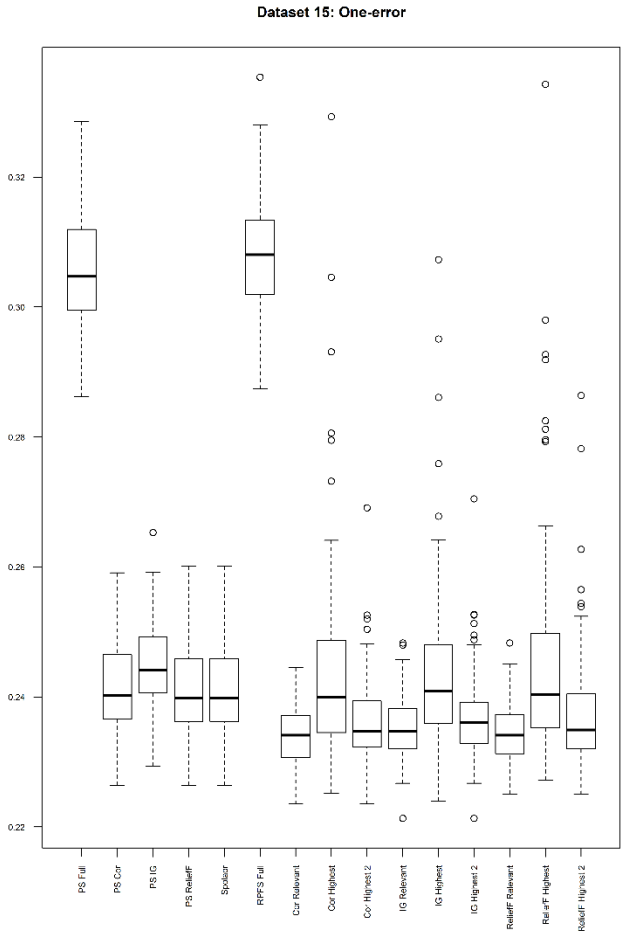


Figure J.58 Dataset 15: One-error.

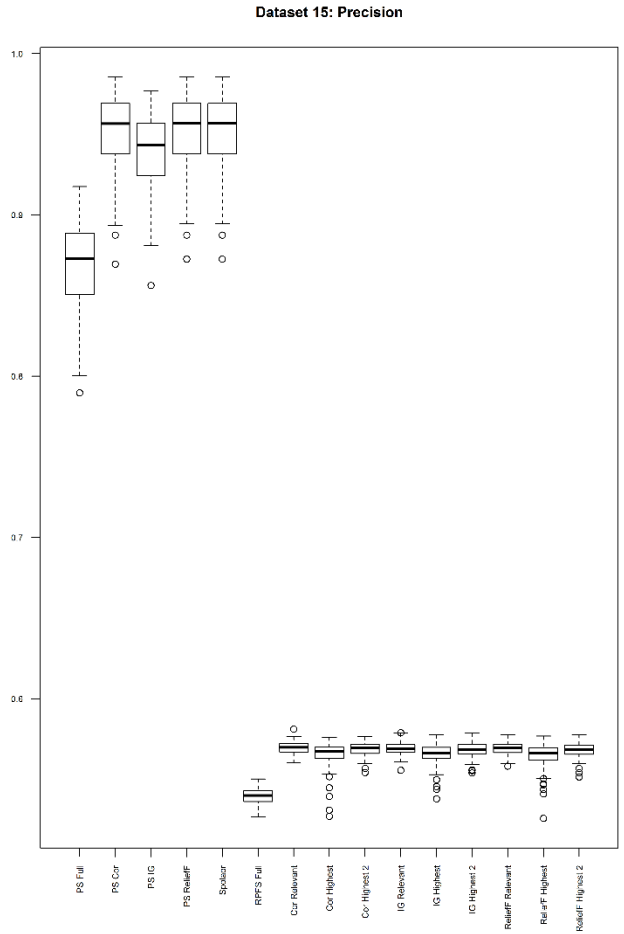


Figure J.59 Dataset 15: Precision.

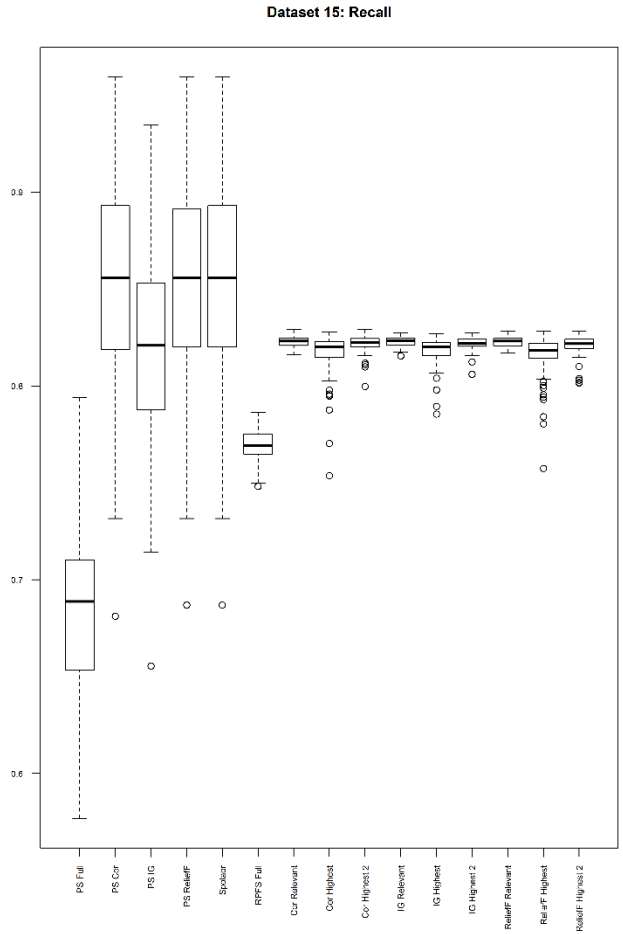


Figure J.60 Dataset 15: Recall.

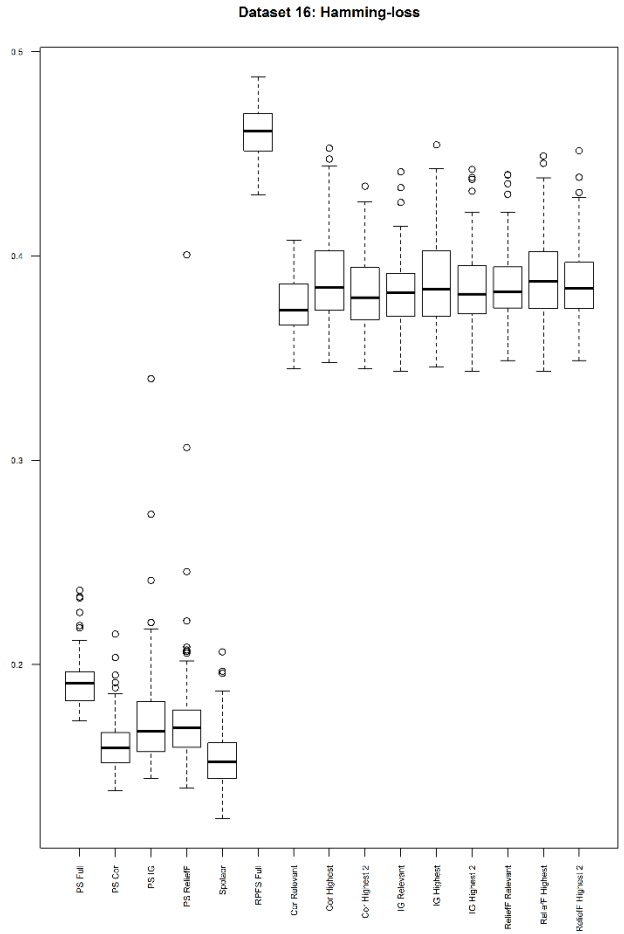


Figure J.61 Dataset 16: Hamming-loss.

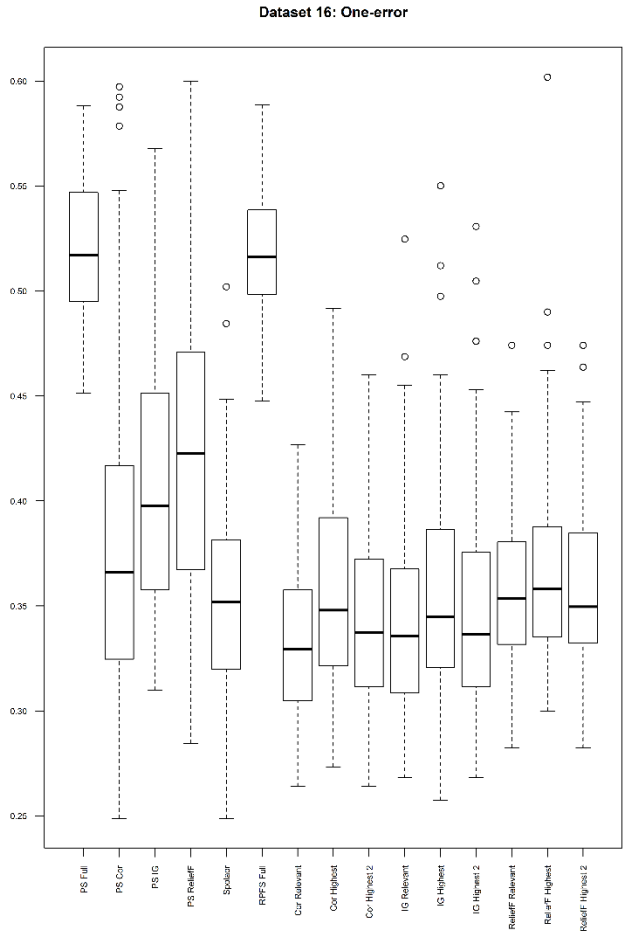


Figure J.62 Dataset 16: One-error.

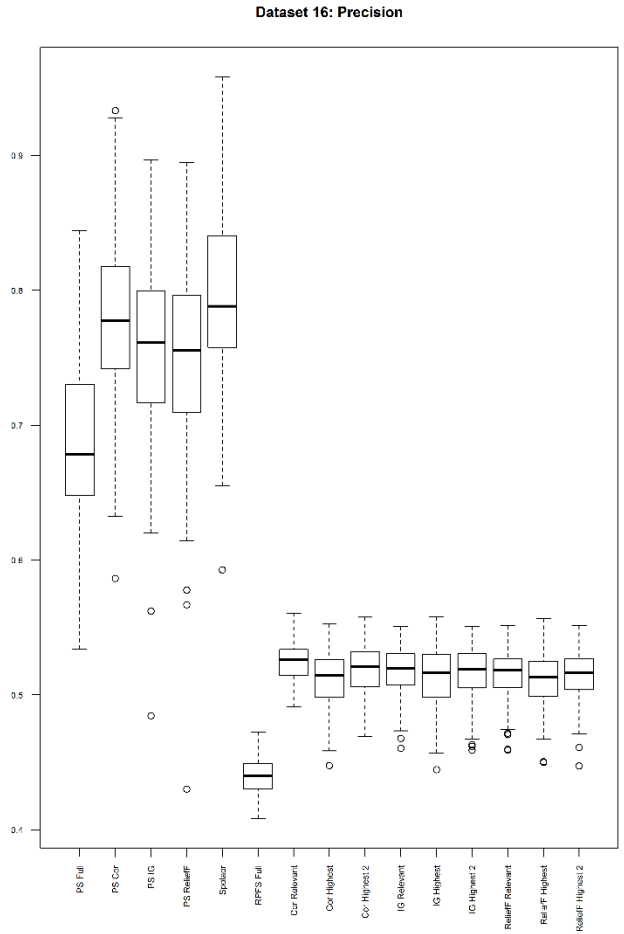


Figure J.63 Dataset 16: Precision.

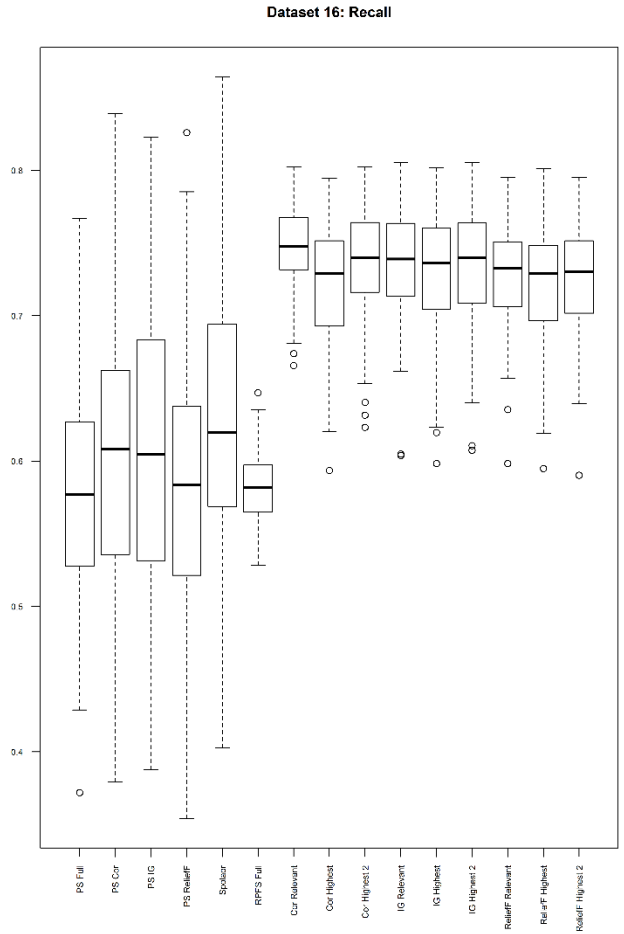


Figure J.64 Dataset 16: Recall.

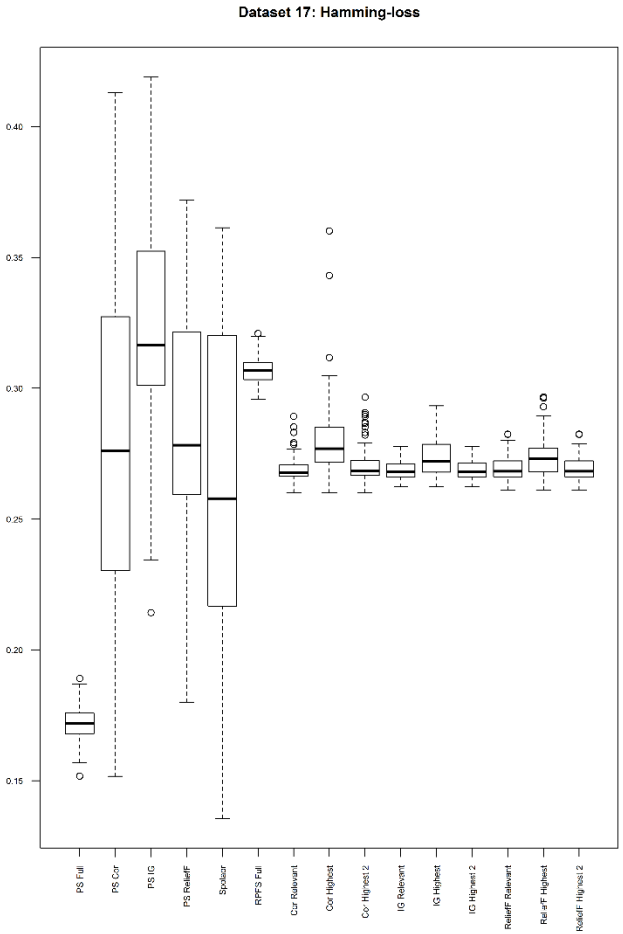


Figure J.65 Dataset 17: Hamming-loss.

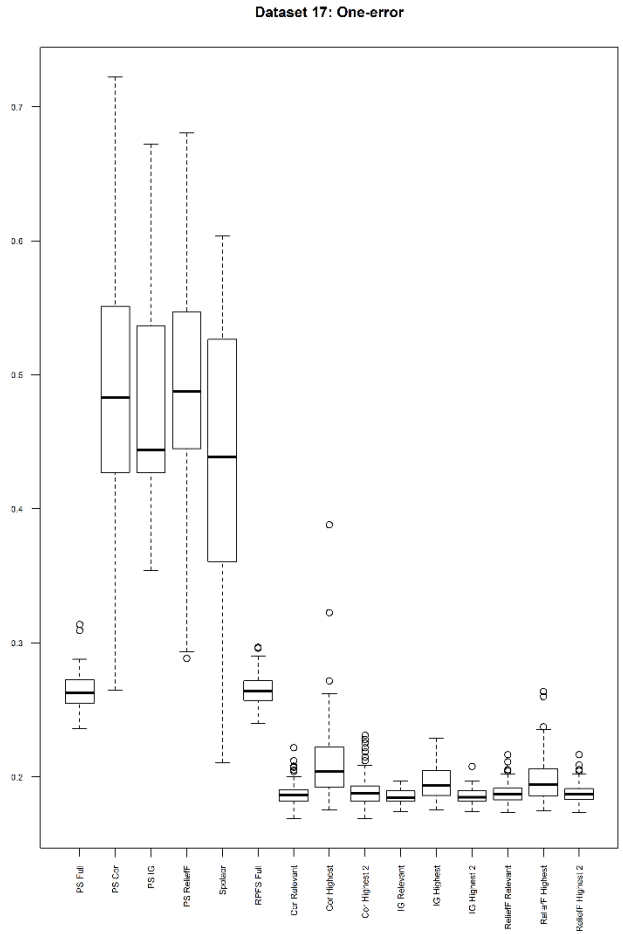


Figure J.66 Dataset 17: One-error.

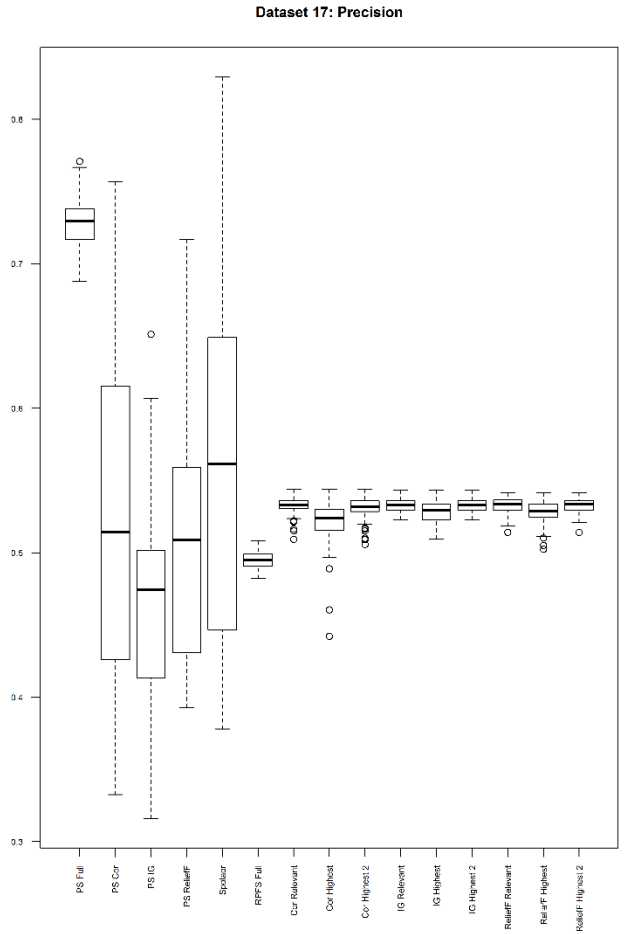


Figure J.67 Dataset 17: Precision.

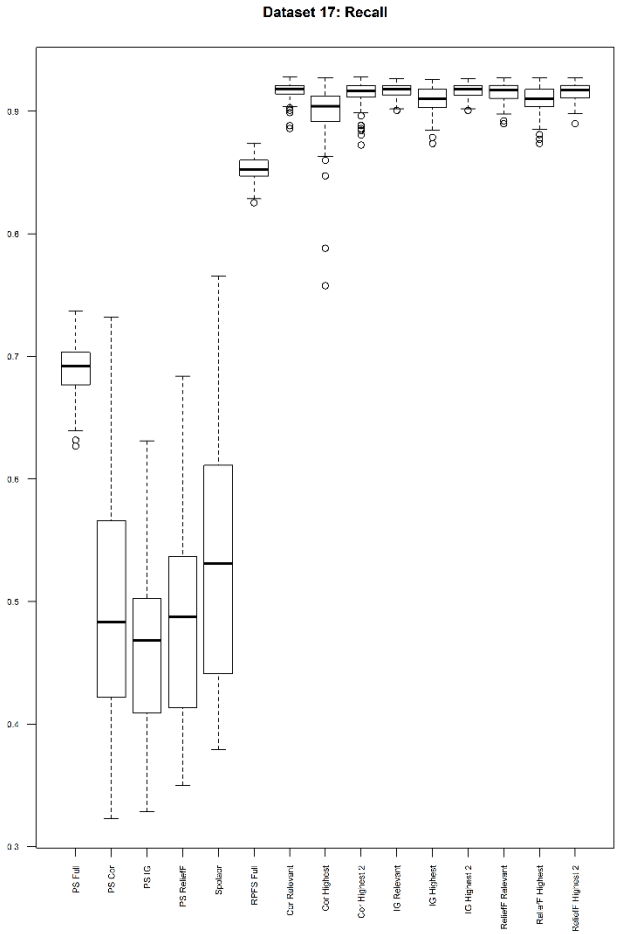


Figure J.68 Dataset 17: Recall.

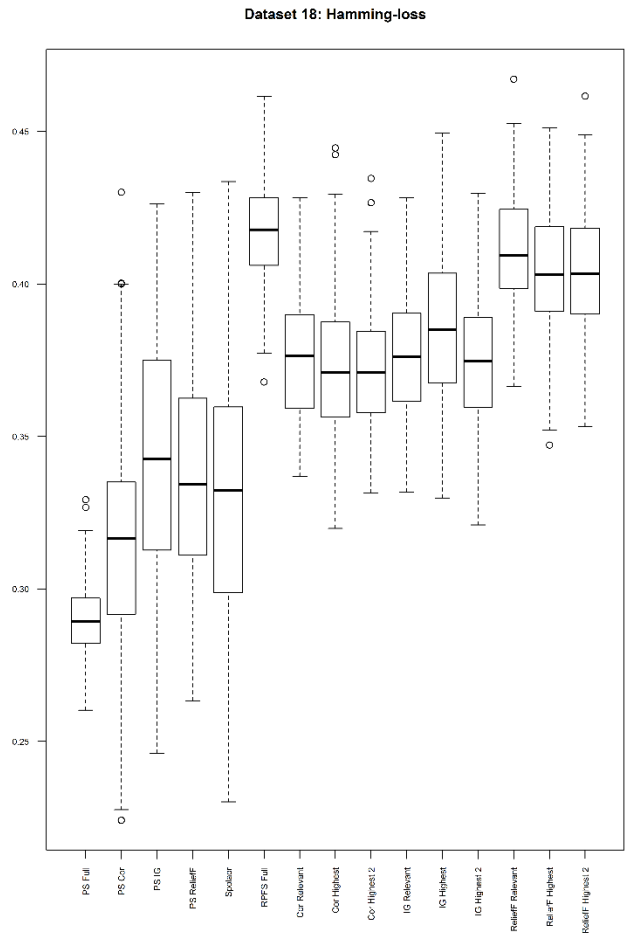


Figure J.69 Dataset 18: Hamming-loss.

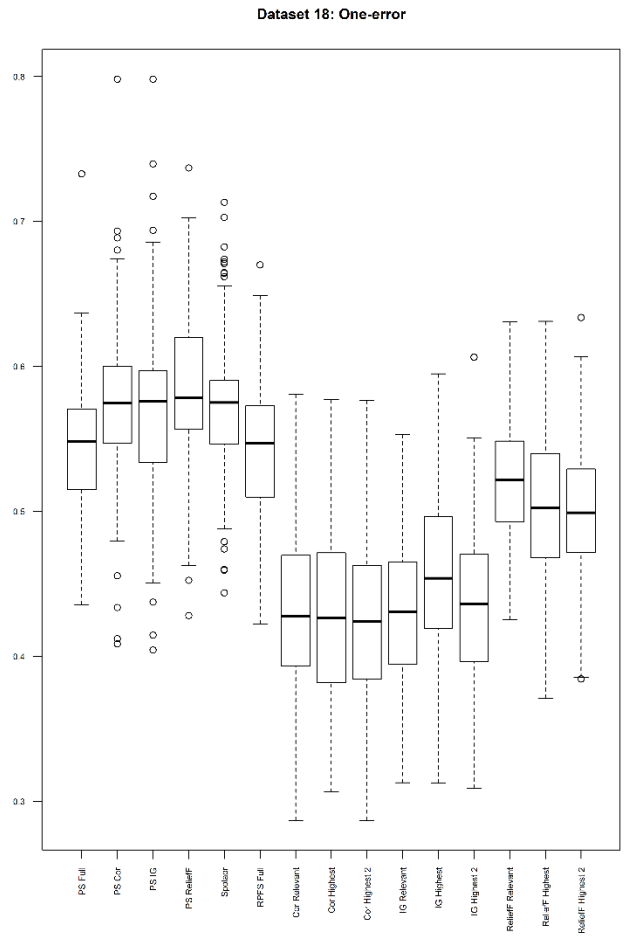


Figure J.70 Dataset 18: One-error.

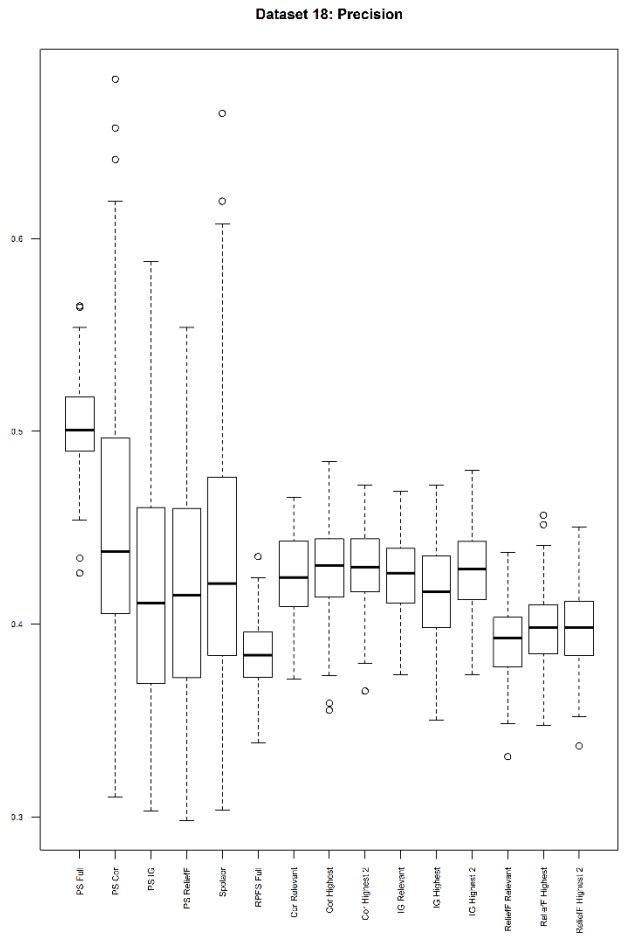


Figure J.71 Dataset 18: Precision.

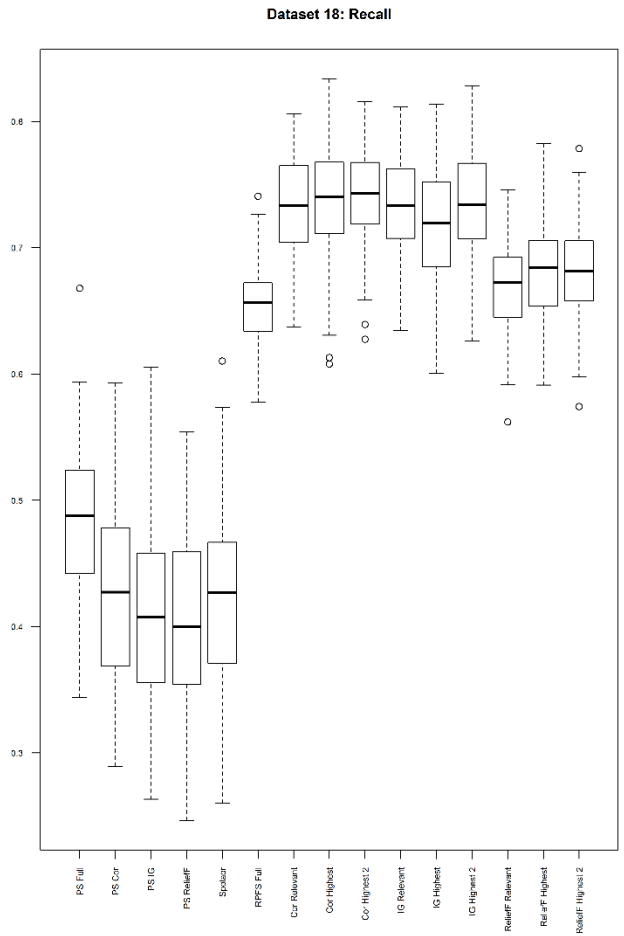


Figure J.72 Dataset 18: Recall.

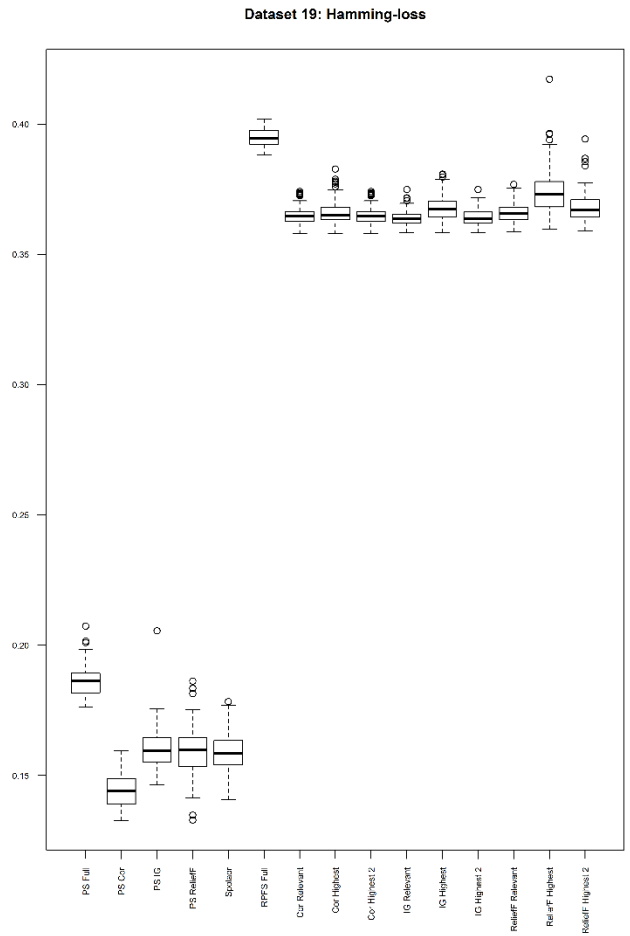


Figure J.73 Dataset 19: Hamming-loss.

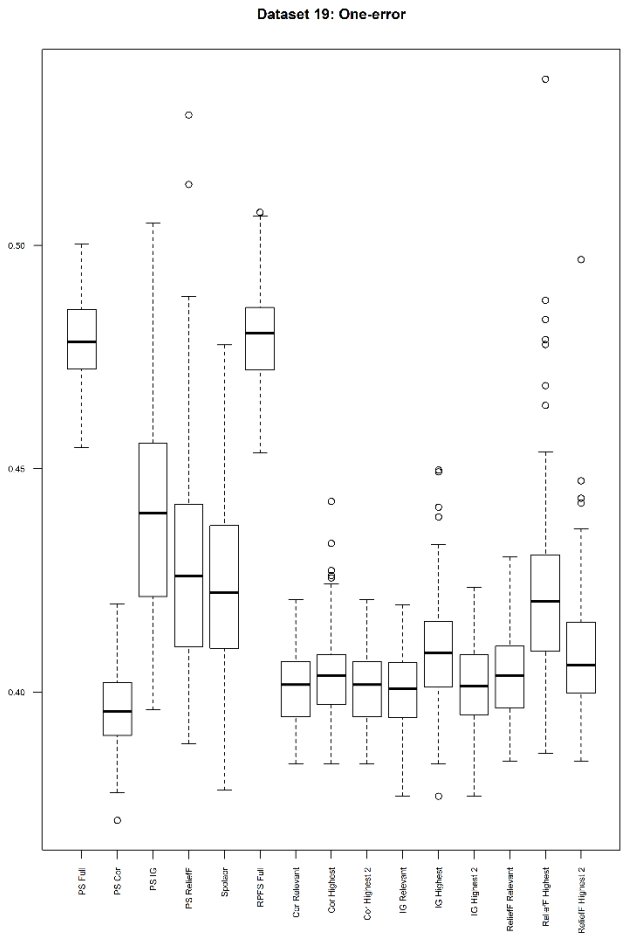


Figure J.74 Dataset 19: One-error.

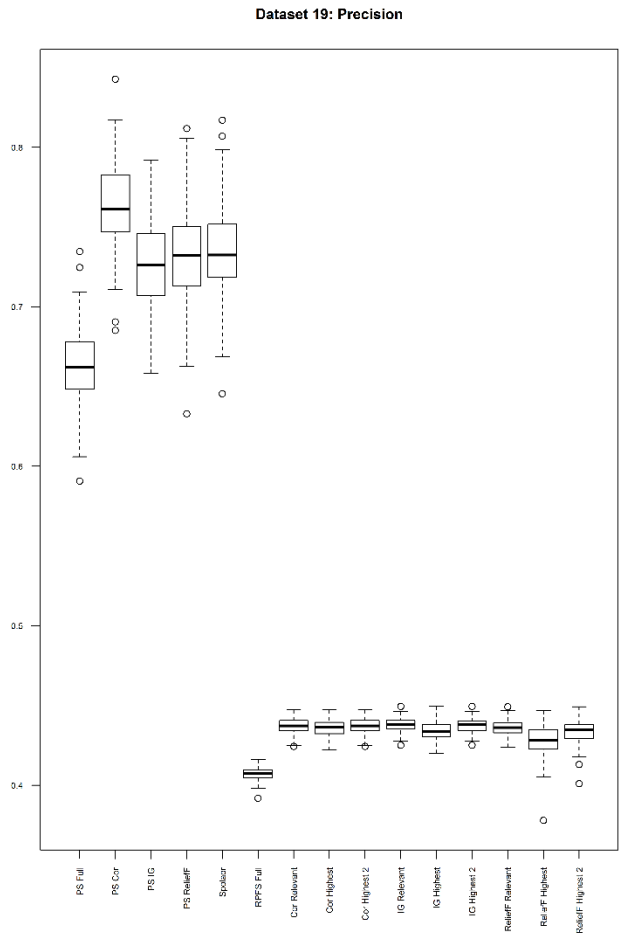


Figure J.75 Dataset 19: Precision.

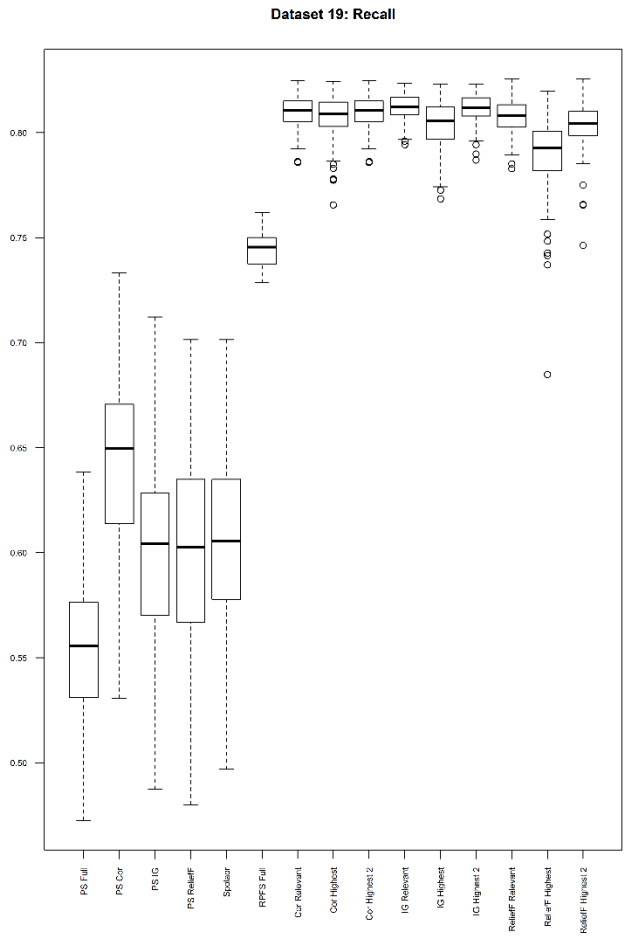


Figure J.76 Dataset 19: Recall.

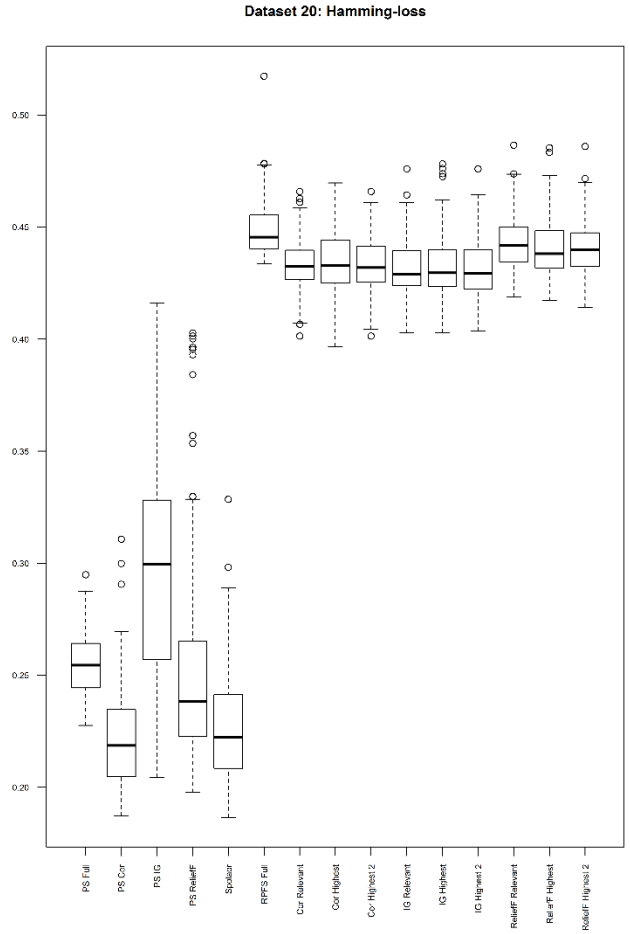


Figure J.77 Dataset 20: Hamming-loss.

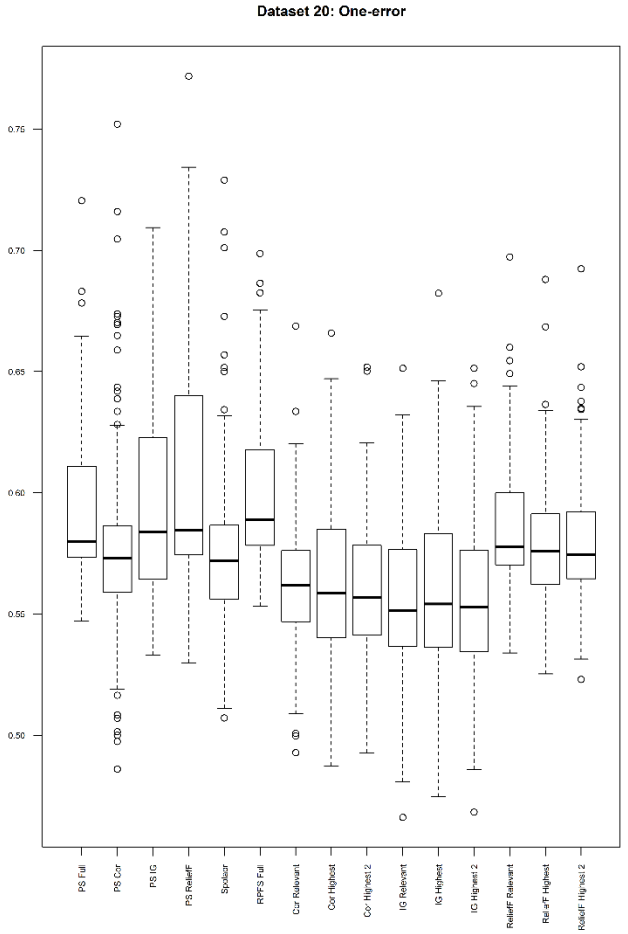


Figure J.78 Dataset 20: One-error.

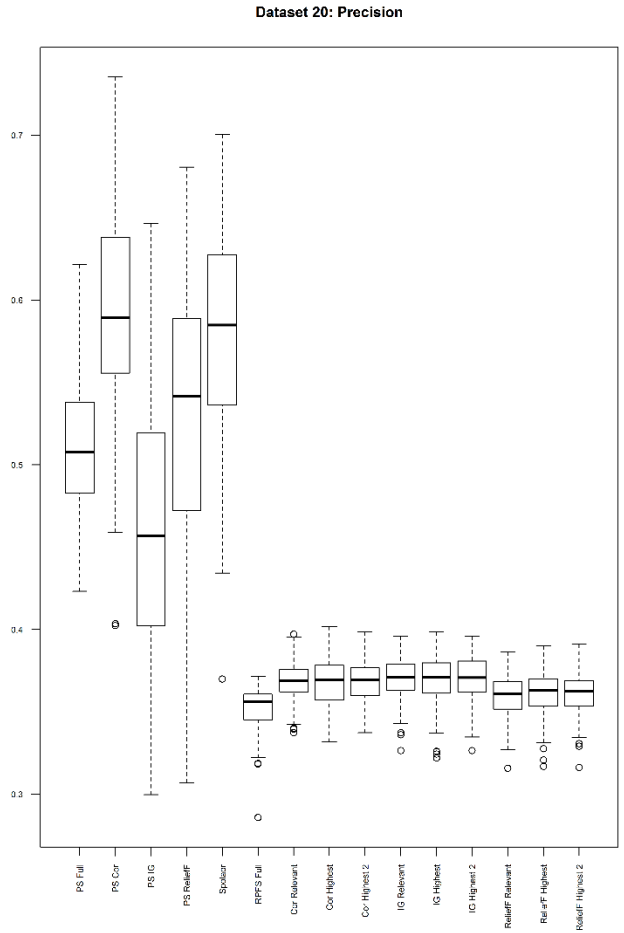


Figure J.79 Dataset 20: Precision.

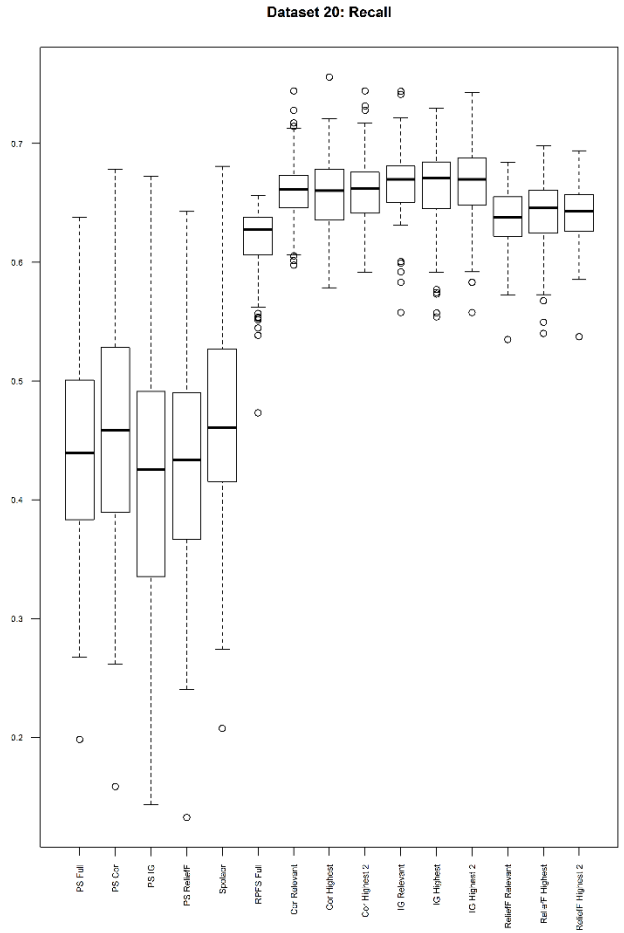


Figure J.80 Dataset 20: Recall.

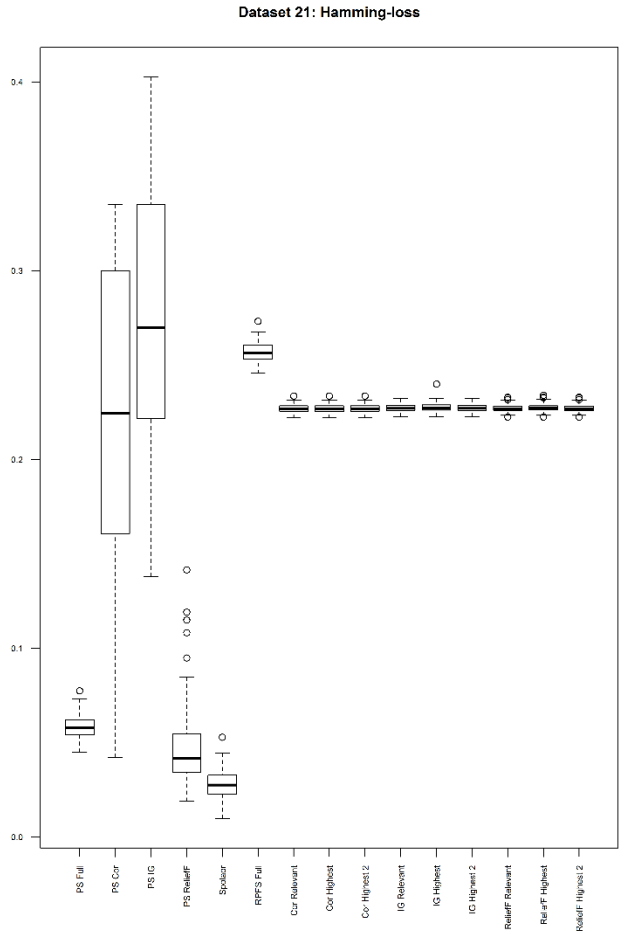


Figure J.81 Dataset 21: Hamming-loss.

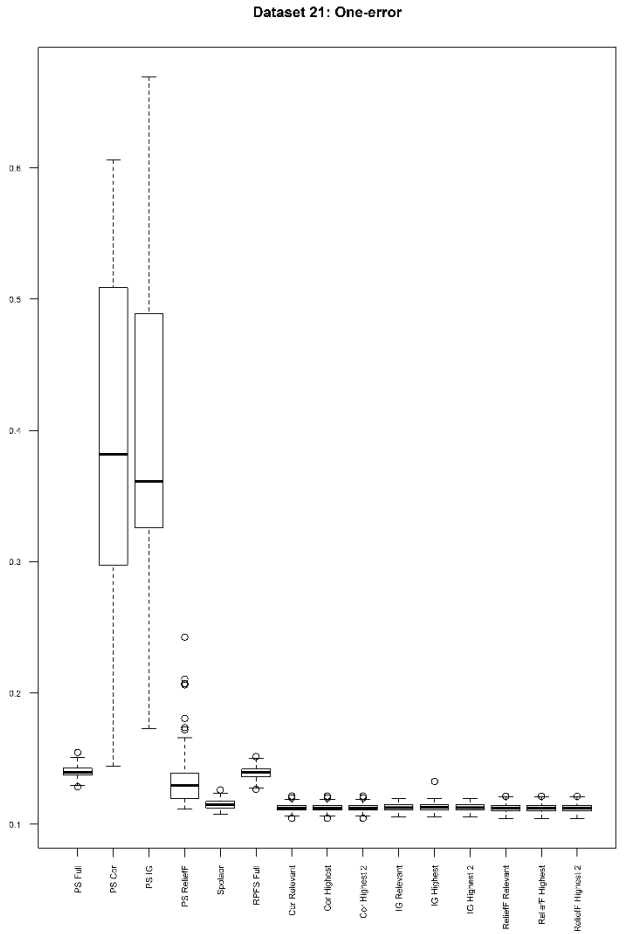


Figure J.82 Dataset 21: One-error.

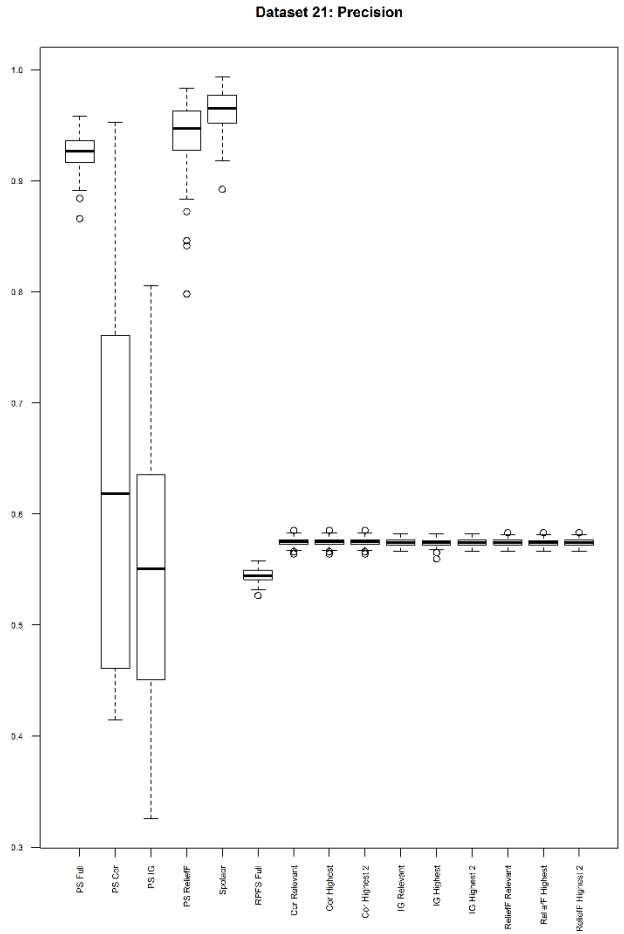


Figure J.83 Dataset 21: Precision.

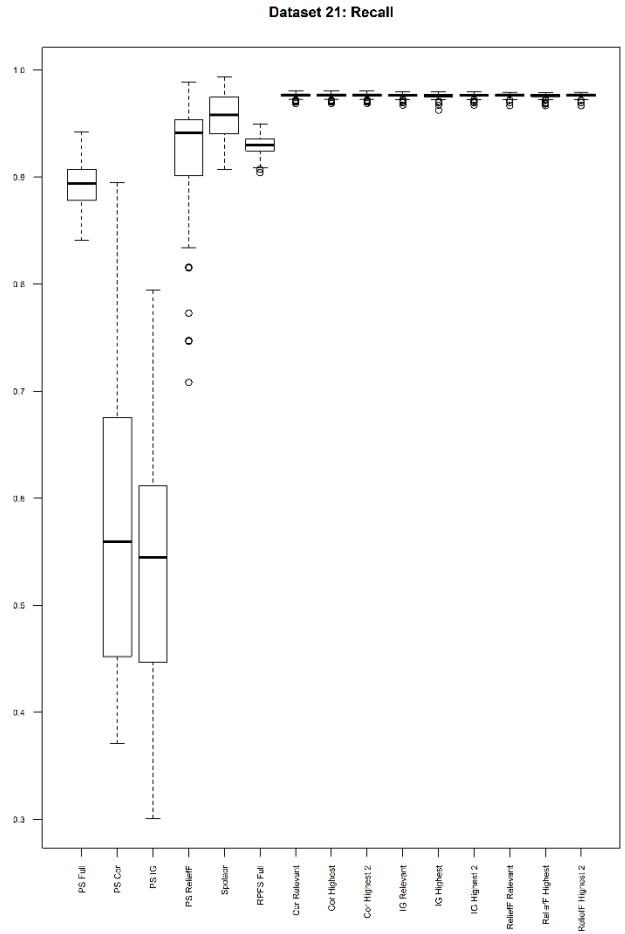


Figure J.84 Dataset 21: Recall.

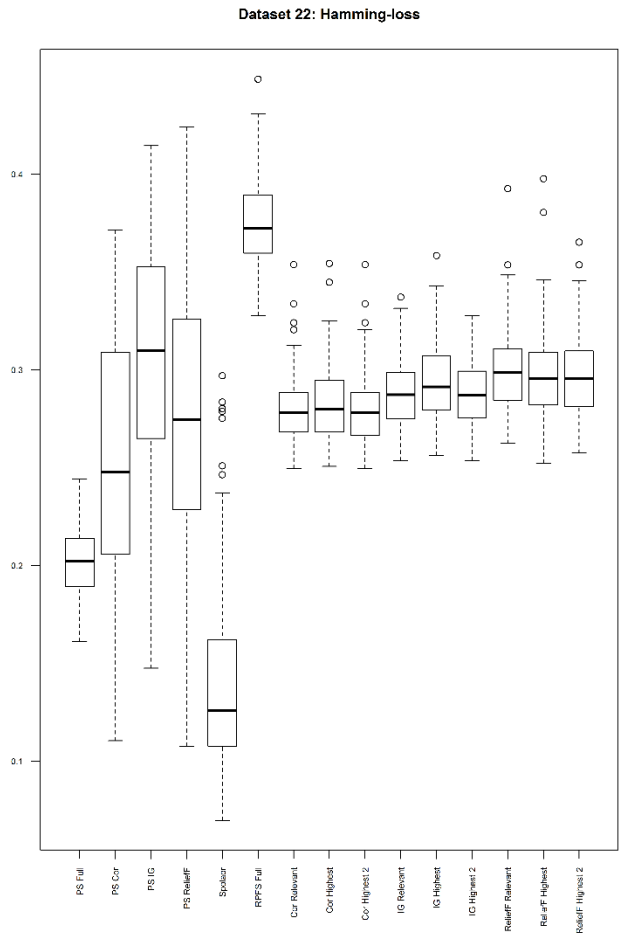


Figure J.85 Dataset 22: Hamming-loss.

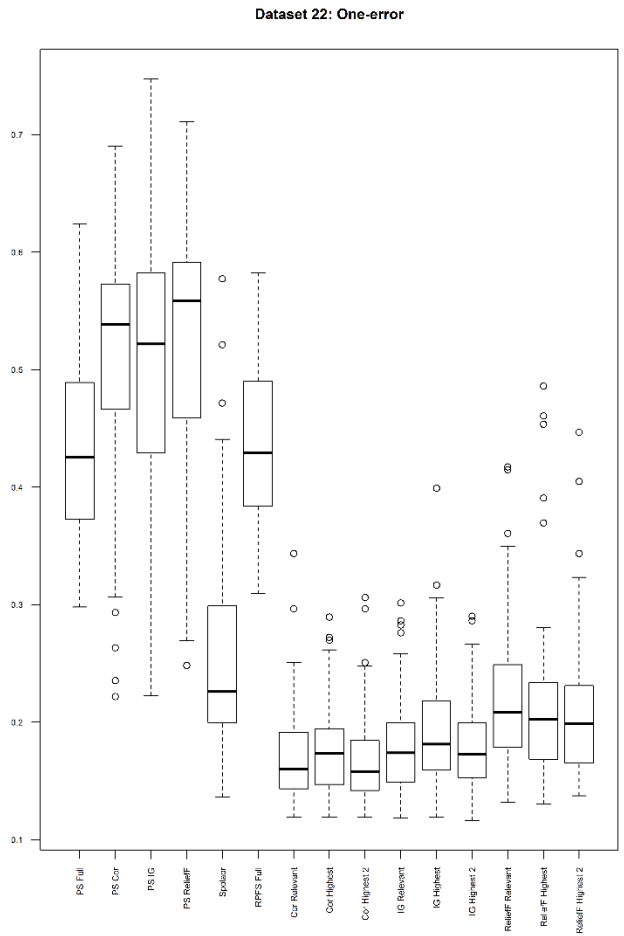


Figure J.86 Dataset 22: One-error.

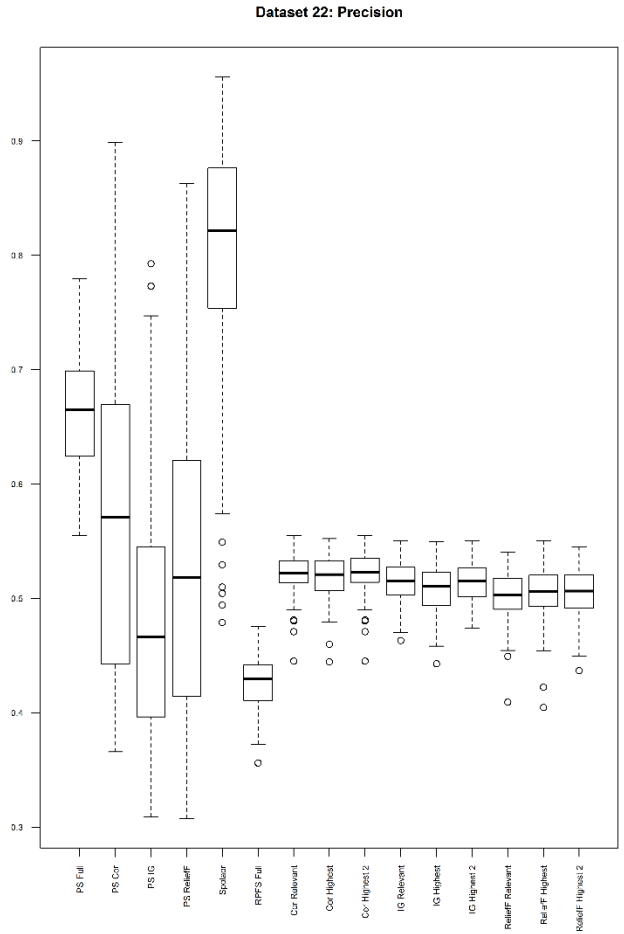


Figure J.87 Dataset 22: Precision.

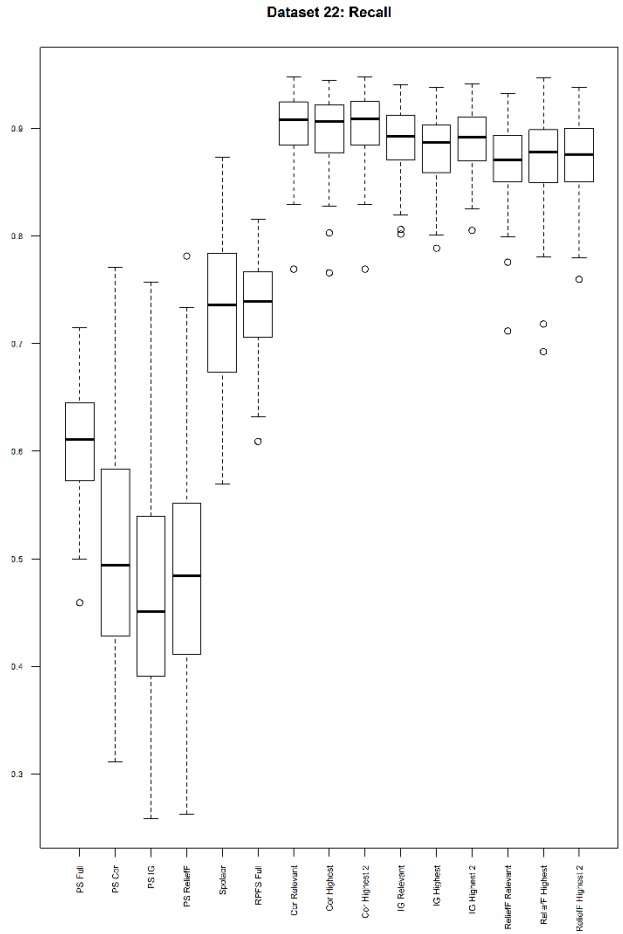


Figure J.88 Dataset 22: Recall.

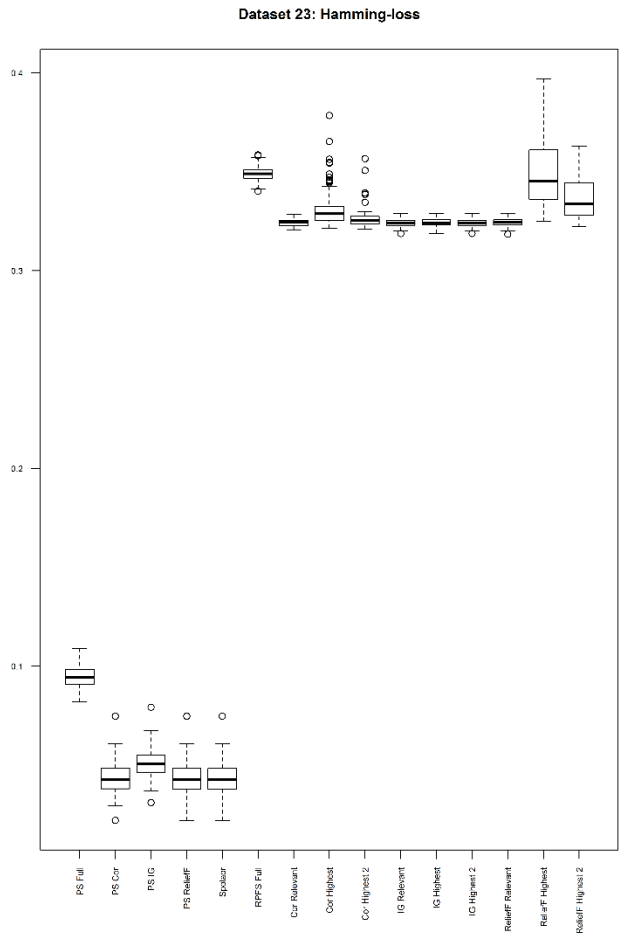


Figure J.89 Dataset 23: Hamming-loss.

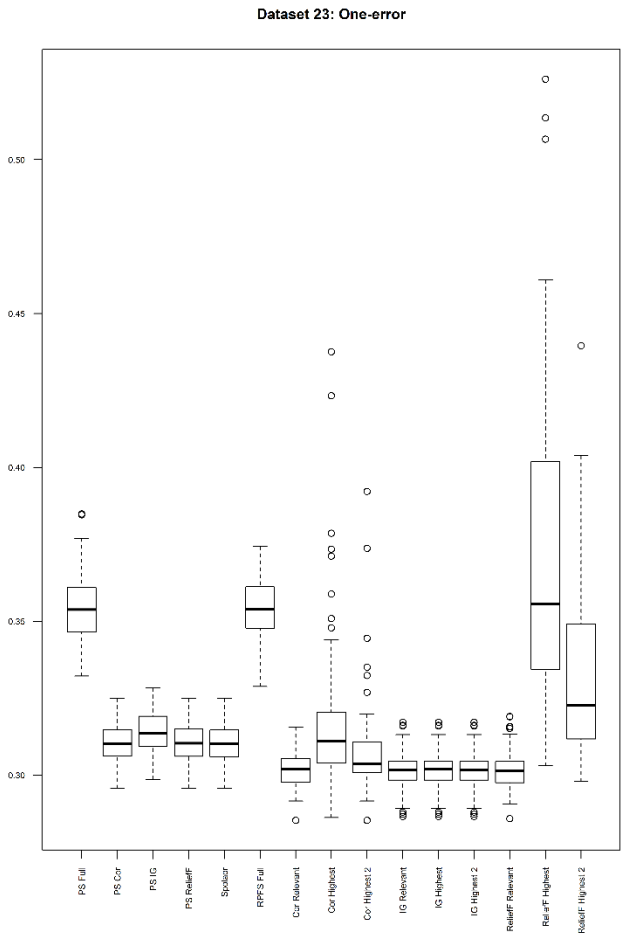


Figure J.90 Dataset 23: One-error.

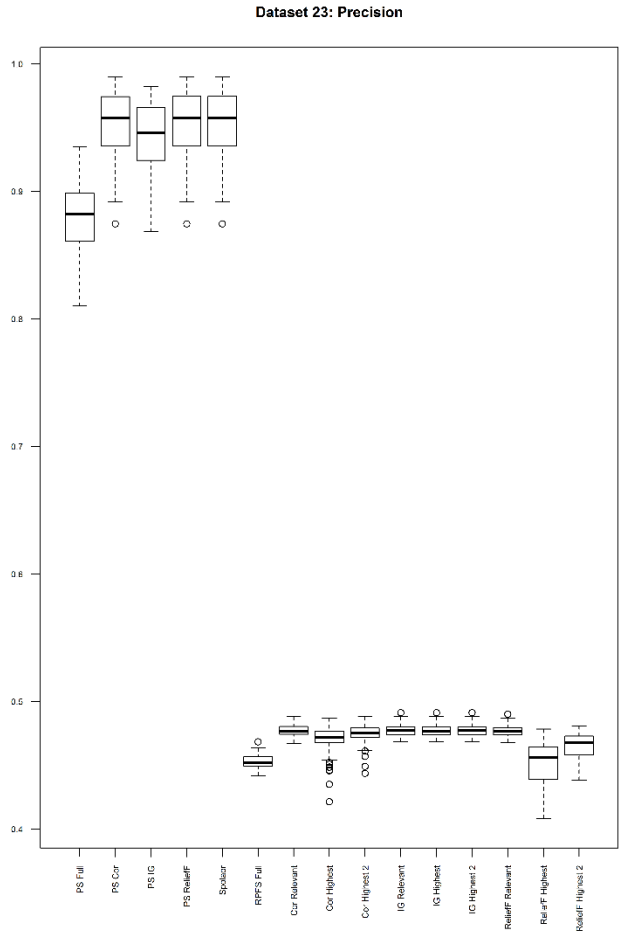


Figure J.91 Dataset 23: Precision.

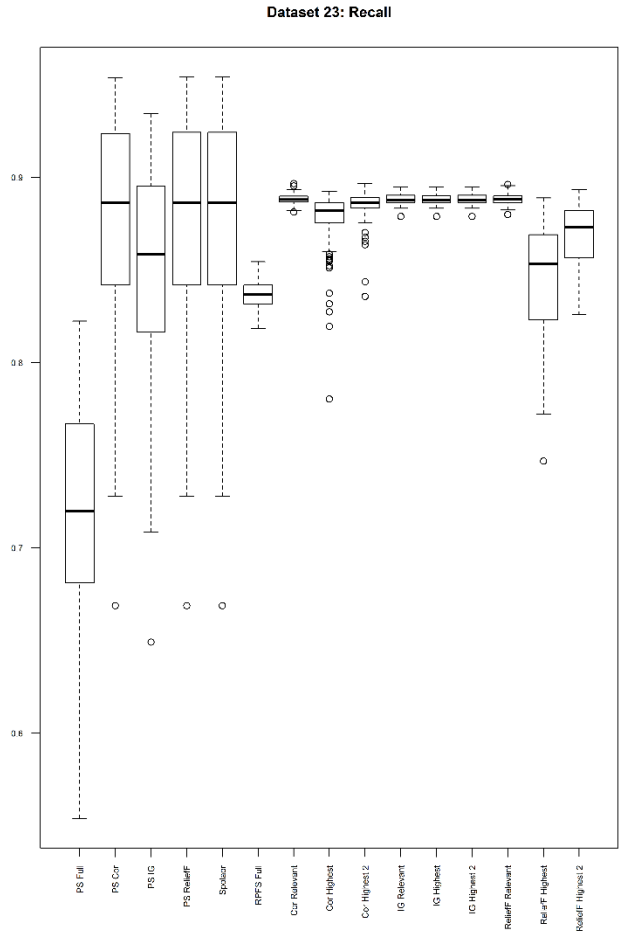


Figure J.92 Dataset 23: Recall.

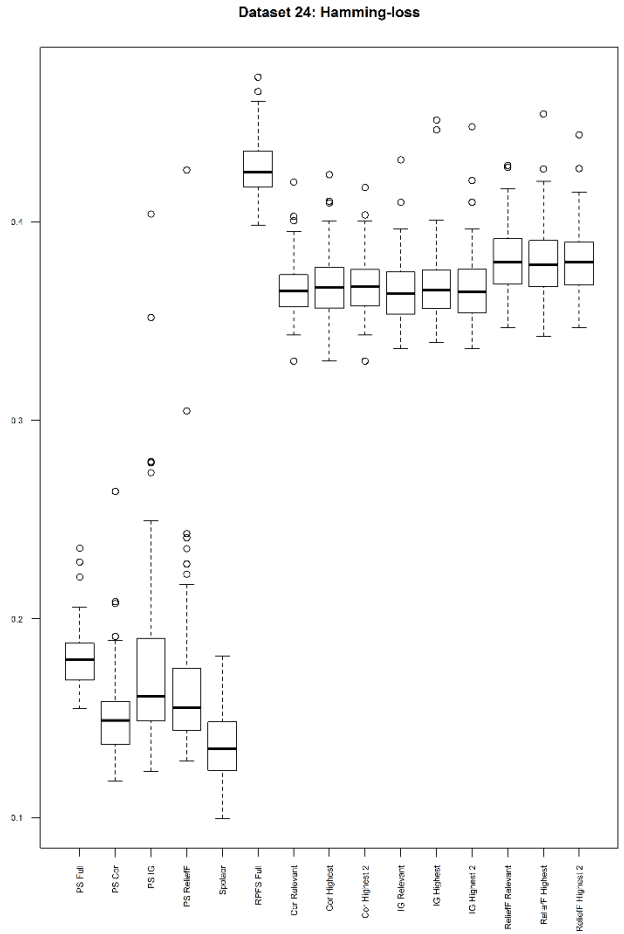


Figure J.93 Dataset 24: Hamming-loss.

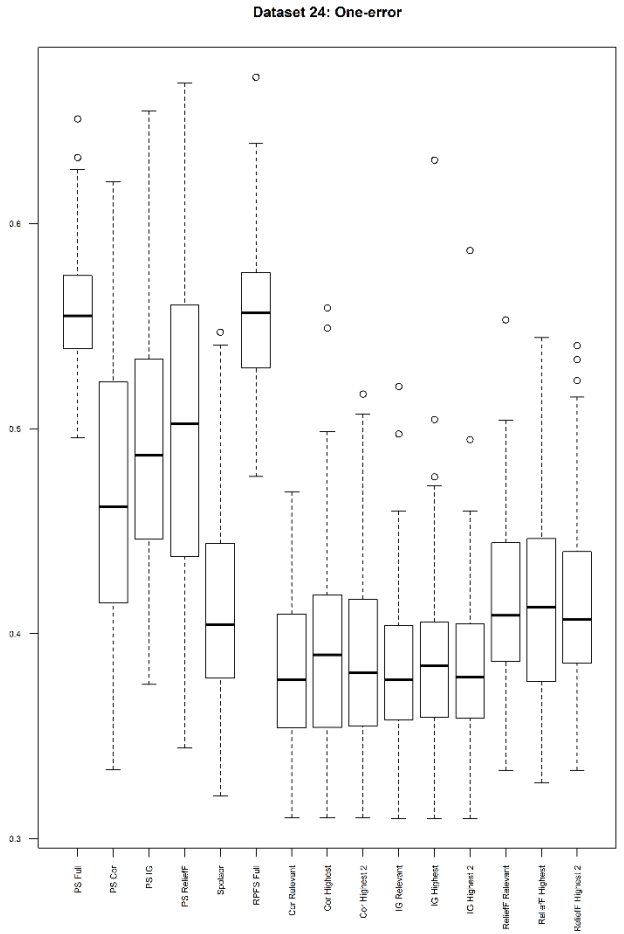


Figure J.94 Dataset 24: One-error.

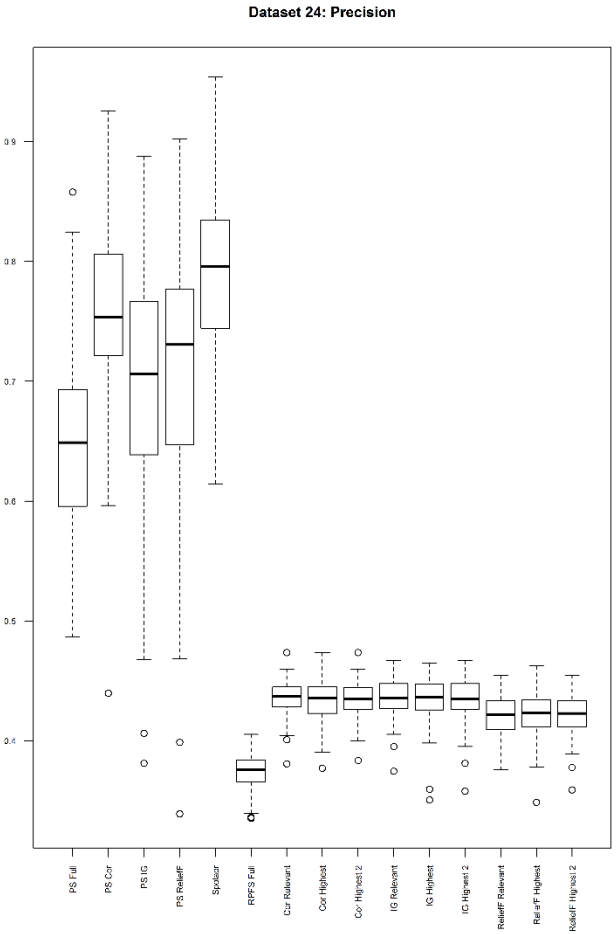


Figure J.95 Dataset 24: Precision.

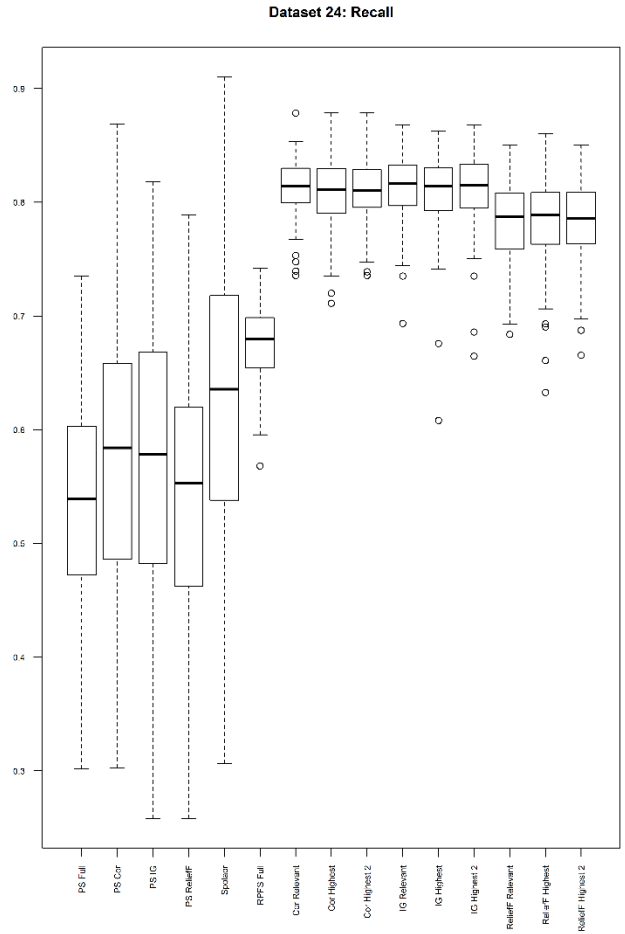


Figure J.96 Dataset 24: Recall.

Appendix K

K.1 Calculating the multi-label evaluation measures

```
function(ylabels,zlabels,rankedlabels)
{
library(gdata)
library(gtools)
library(gplots)
library(gmodels)

#########################################################################
# This program computes various measures of classification accuracy.    #
# The input to the program is:                                          #
#     1. ylabels: an indicator matrix containing the true labels for    #
#           a set of Nnew new cases.                                    #
#     2. zlabels: an indicator matrix containing the predicted labels   #
#           for the Nnew new cases.                                     #
#     3. rankedlabels: the structure of this matrix is as follows:      #
#           in each row we have the ranks 1,2,...,q, with rank 1        #
#           signifying the most relevant label, etc.                    #
#########################################################################

#########################################################################
# We first do some bookkeeping.                                         #
#########################################################################

ylabels = as.matrix(ylabels)
zlabels = as.matrix(zlabels)
rankedlabels = as.matrix(rankedlabels)
Nnew = nrow(ylabels)
q = ncol(ylabels)
```

```
#############################################################################
# The following measures are computed:                                      #
# Example-based measures:                                                   #
#     1. Hamming loss (Hloss)                                               #
#     2. Classification accuracy (clacc)                                    #
#     3. Precision (precision)                                              #
#     4. Recall (recall)                                                    #
#     5. F-one, first version (F11)                                         #
#     6. F-one, second version (F12)                                        #
#     7. Accuracy (accuracy)                                                #
# Ranking-based measures:                                                   #
#     1. One-error (one.error)                                              #
#     2. Coverage (coverage)                                                #
#     3. Ranking loss (ranking.loss)                                        #
#     4. Average precision (aveprecision)                                   #
#############################################################################

#############################################################################
# First, compute the example-based measures.                                #
#############################################################################

yminz = ylabels-zlabels
ymaalz = ylabels*zlabels
ydeltaz = apply(yminz,1,function(x) sum(abs(x)))
nylabels = apply(ylabels,1,sum)
nzlabels = apply(zlabels,1,sum)
somnylablesnzlabels = (nylabels + nzlabels)
proportion1 = ydeltaz/q

Hloss = mean(proportion1)
clacc = sum(ydeltaz==0)/Nnew

yinterseksiez = apply(ymaalz,1,sum)
yverenigz = nylabels+nzlabels-yinterseksiez
proportion2 = yinterseksiez[nzlabels>0]/nzlabels[nzlabels>0]
proportion3 = yinterseksiez[nylabels>0]/nylabels[nylabels>0]
#proportion4 = yinterseksiez/(nylabels+nzlabels)
```

```
proportion4 =
      yinterseksiez[somnylablesnzlabels>0]/somnylablesnzlabels[somnylablesnzl
      abels>0]
proportion5 = yinterseksiez[nylabels>0]/yverenigz[nylabels>0]

precision = mean(proportion2)
recall = mean(proportion3)
F11 = 2*mean(proportion4)
F12 = 1/mean(c(1/precision,1/recall))
accuracy = mean(proportion5)

#############################################################################
# Now, compute the ranking-based measures.                                  #
#############################################################################

ylabtimesrank = ylabels*rankedlabels
nie.ylabtimesrank = (matrix(1,nrow=Nnew,ncol=q)-ylabels)*rankedlabels

one.error = mean(apply(ylabtimesrank,1,function(x) min(x[x>0]))!=1)
coverage =
      mean(apply(ylabtimesrank[nylabels>0,],1,max)/apply(ylabels[nylabels>0,]
      ,1,sum))-1

ranking.fn = function(j) {
      indeks1 = which(ylabels[j,]==1)
      indeks2 = which(ylabels[j,]==0)
      som = 0
      for (i in 1:length(indeks1))
            som=som+as.numeric(ylabtimesrank[j,indeks1[i]]>nie.ylabtimesrank[
            j,indeks2])
            sum(som)/(sum(ylabels[j,])*(q-sum(ylabels[j,])))
}

ranking = rep(0,Nnew)
for (j in 1:Nnew) if((sum(ylabels[j,]>0))&((q-sum(ylabels[j,]))>0))
      {ranking[j] = ranking.fn(j)
      ranking.loss = mean(ranking)
}
```

```
precision.fn = function(j) {
      indeks1 = which(ylabels[j,]==1)
      som = 0
      for (i in 1:length(indeks1)){
            som1=0
            for (k in 1:length(indeks1)){
                  som1=som1+as.numeric(ylabtimesrank[j,indeks1[k]]<=ylabtimes
                  rank[j,indeks1[i]])}
            som=som+som1/ylabtimesrank[j,indeks1[i]]}
      som/length(indeks1)
      }

rank.precision = rep(0,Nnew)
for (j in 1:Nnew) if(nylabels[j]>0)
      {rank.precision[j] = precision.fn(j)
      aveprecision = mean(rank.precision)
}

afvoer =
list(Hloss,clacc,precision,recall,F11,F12,accuracy,one.error,coverage,ranking
.loss,aveprecision)
return(afvoer)
}
```

K.2 Implementation of RPFS procedure based on ReliefF using XGBoost classifier for the Emotions dataset

```
function (p,alpha,mrep,n.splits)
{
#############################################################################
# For Emotions data set                                                     #
# p refers to the number of features (= 72)                                 #
# q refers to the number of labels (= 6)                                    #
# k refers to the number of relevant features (= 72-)                       #
# N refers to the number of instances                                       #
# alpha and mrep are used to threshold the W-values obtained using          #
#     ReliefF (alpha = 0.05 & mrep = 10000)                                 #
#############################################################################
```

```
#########################################################################
# Step 1:   Load libraries and read in data                             #
#########################################################################

library(foreign)
library(miscFuncs)
library(CORElearn)
library(UBbipl)
library(UBFigs)
library(ggplot2)
library(xgboost)

xydata = read.arff("C:\\...\\emotions.arff")
ydata=matrix(as.numeric(unlist(xydata[,(p+1):ncol(xydata)])),nrow=nrow(xydata
      ),ncol=ncol(xydata) - p,byrow=FALSE)-1
xdata=matrix(as.numeric(unlist(xydata[,1:p])),nrow=nrow(xydata),ncol=p,byrow=
      FALSE)
xydata=cbind(xdata,ydata)
N = nrow(xydata)
q = ncol(xydata) - p

#########################################################################
# Initialise the following:                                             #
# 1. numberofgroups.mat = n.splits x 1 matrix that captures the number  #
#     of groups used for each split                                     #
# 2. results.full, results.k, results.best and results.best2 =          #
#     n.splits x 11 matrices that contains the evaluation measures for  #
#     each split                                                        #
# 3. irrelevant.feat.mat = n.splits x p matrix that captures the        #
#     irrelevant features for each n.splits                             #
# 4. best.mat = n.splits x p matrix that captures the feature with the  #
#     highest rank in each group                                        #
# 5. best2.mat = n.splits x p matrix that captures the two feature with #
#     the two highest ranks in each group                               #
#########################################################################

numberofgroups.mat = matrix(0, nrow = n.splits, ncol = 1)
results.full = matrix(0,nrow = n.splits, ncol = 11)
```

```
results.k = matrix(0,nrow = n.splits, ncol = 11)
results.best = matrix(0,nrow = n.splits, ncol = 11)
results.best2 = matrix(0,nrow = n.splits, ncol = 11)
irrelevant.feat.mat = matrix(0,nrow = n.splits, ncol = p)
best.mat = matrix(0,nrow = n.splits, ncol = p)
best2.mat = matrix(0,nrow = n.splits, ncol = p)
mintesterror = matrix(0, nrow = n.splits, ncol = q)
ntreesvec = matrix(0, nrow = n.splits, ncol = q)
eta = matrix(0, nrow = n.splits, ncol = q)
depth = matrix(0, nrow = n.splits, ncol = q)

#############################################################################
# The MC loop now starts.                                                   #
#############################################################################

for (jjj in 1:n.splits) {

#############################################################################
# Set up training and test data sets.                                       #
# We use 70% of the data for training and 30% for testing.                  #
#############################################################################

      print(jjj)

      xydata = as.matrix(xydata)
      trainindekse = sample(1:N,N*0.7,replace=FALSE)
      xydatatrain = as.matrix(xydata[trainindekse,])
      xydatatest = xydata[-trainindekse,]
      Ntrain = nrow(xydatatrain)
      xdatatrain = xydatatrain[,1:p]
      ydatatrain = xydatatrain[,(p+1):ncol(xydatatrain)]
      xdatatest = xydatatest[,1:p]
      ydatatest = xydatatest[,(p+1):ncol(xydatatest)]
      Nnew = nrow(xydatatest)
```

```
#########################################################################
# Step 2:   Find relevant features and set-up relevance matrix          #
#                 ReliefF                                               #
#########################################################################

      Wvalues = matrix(rep(0),ncol = q,nrow = p)
      q = ncol(ydatatrain)
      p = ncol(xdatatrain)
      for (i in 1:q)    {
            yvec = as.factor(ydatatrain[,i])
            datatrain = data.frame(yvec,xdatatrain)
            value =
                  attrEval(as.factor(yvec)~.,datatrain,estimator="ReliefFexpR
                  ank",ReliefIterations = mrep)
            for (j in 1:p) Wvalues[j,i]=value[j]
      }
      tauthresh = 1/sqrt(alpha*mrep)
      reliefFrelmat = matrix(0,nrow=p,ncol=q)
      reliefFrelmat[Wvalues>tauthresh]=1
      write.table(reliefFrelmat,"C:\\...\\EmotionsM2relmat.txt")
      write.table(Wvalues,"C:\\...\\EmotionsM2Wvalue.txt")


#########################################################################
# Output:                                                               #
# relmat = pxq relevance matrix                                         #
# Wvalue = W values obtained from ReliefF                               #
#########################################################################


#########################################################################
# Step 3:   Create coordinates for MCA biplot representing groups       #
#           Create groups based on MCA                                  #
#########################################################################

      relmat = read.table("C:\\...\\EmotionsM2relmat.txt")
      Z.02 = MCABIPL(relmat)$Z.0
      Z2 = MCABIPL(relmat)$Z
```

```
      reltemp2 = Z.02
      p = nrow(reltemp2)
      q = ncol(reltemp2)
      deletedrow = matrix(-1,nrow = 1, ncol = q)
      groupmat = matrix(0, nrow = p, ncol = p)
      uniquerows = matrix(0, nrow = p, ncol = q)
      ngroup = 0
      maxningroup = 0
      for (count1 in 1:p) {
            thisgroup = reltemp2[count1,]
            if (sum(thisgroup == deletedrow) < q)     {
                  ngroup = ngroup + 1
                  uniquerows[ngroup,] = as.matrix(thisgroup)
                  groupmat[ngroup,1] = count1
                  ningroup = 1
                  if (count1 < p)
                        for (count2 in (count1+1):p)
                        if (sum(reltemp2[count2,] == thisgroup) == q) {
                              ningroup = ningroup + 1
                              maxningroup = max(maxningroup,ningroup)
                              groupmat[ngroup, ningroup] = count2
                              reltemp2[count2,] = as.matrix(deletedrow)
                        }
                  }
            }
      groupmatout = groupmat[1:ngroup,1:maxningroup]
      uniquerowsout = uniquerows[1:ngroup,]
      groupsize = rowSums(groupmatout!=0)
      numberofgroups = nrow(groupmatout)
      numberofgroups.mat[jjj,1] = numberofgroups

#########################################################################
# Step 4: Perform XGBoost on the full set of p features                 #
#########################################################################

      Hloss.mat = matrix(0,nrow = 1, ncol = 1)
      clacc.mat = matrix(0, nrow = 1, ncol = 1)
      prec.mat = matrix(0, nrow = 1, ncol = 1)
```

```
recall.mat = matrix(0, nrow = 1, ncol = 1)
F11.mat = matrix(0, nrow = 1, ncol = 1)
F12.mat = matrix(0, nrow = 1, ncol = 1)
accuracy.mat = matrix(0, nrow = 1, ncol = 1)
one.error.mat = matrix(0, nrow = 1, ncol = 1)
coverage.mat = matrix(0, nrow = 1, ncol = 1)
rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
aveprecision.mat = matrix(0, nrow = 1, ncol = 1)
Ntrain = nrow(xydatatrain)
Ntest = nrow(xydatatest)
postprob = matrix(0, nrow = Ntest, ncol = q)
Zmatind = matrix(0, nrow = Ntest, ncol = q)
Zmatranks = matrix(0, nrow = Ntest, ncol = q)
testerrorarray = array(0, dim = c(3,3,q))
ntreesarray = array(0,dim = c(3,3,q))
Maxdepth = matrix(0,nrow = 3,ncol = 3)
etamat = matrix(0,nrow = 3, ncol = 3)
Maxdepth[,1] = 1
Maxdepth[,2] = 2
Maxdepth[,3] = 4
etamat[1,] = 0.3
etamat[2,] = 0.5
etamat[3,] = 0.7

for (m in 1:q)    {
      for (j in 1:3) {
            for (i in 1:3)    {
                  cv = xgb.cv(data = as.matrix(xdatatrain), label =
                        ydatatrain[,m], nrounds = 100, nfold = 5,
                        objective = "binary:logistic", eta =
                        etamat[i,j], max_depth = Maxdepth[i,j],
                        early_stopping_rounds = NULL, verbose = 0)
                  ntrees = which.min(cv$evaluation_log$test_error_mean)
                  testerror = min(cv$evaluation_log$test_error_mean)
                  testerrorarray[i,j,m] = testerror
                  ntreesarray[i,j,m] = ntrees
            }
      }
```

```
      mintesterror[jjj,m] = min(testerrorarray[,,m])
      ind = which(testerrorarray[,,m] == min(testerrorarray[,,m]),
            arr.ind = TRUE)
      ntreesvec[jjj,m] = ntreesarray[ind[[1]],ind[[2]],m]
      eta[jjj,m] = etamat[ind[[1]],ind[[2]]]
      depth[jjj,m] = Maxdepth[ind[[1]],ind[[2]]]
}

write.table(mintesterror,"C:\\...\\EmotionsMinTestError.txt")
write.table(ntreesvec,"C:\\...\\EmotionsNumberofTrees.txt")
write.table(depth,"C:\\...\\EmotionsDepth.txt")
write.table(eta,"C:\\...\\EmotionsEta.txt")

for (m in 1:q)    {
      model_xgb = xgboost(data = as.matrix(xdatatrain), label =
            ydatatrain[,m], nrounds = ntreesvec[jjj,m], objective =
            "binary:logistic", eta = eta[jjj,m], depth = depth[jjj,m],
            verbose = 0)
      postprob[,m] = predict(model_xgb, newdata = as.matrix(xdatatest))
}

Zmatind[postprob>0.5]=1
for (i in 1:Ntest)      {
      Zmatranks[i,] = rank(-postprob[i,])
}
measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
Hloss.mat = measures[[1]]
clacc.mat = measures[[2]]
prec.mat = measures[[3]]
recall.mat = measures[[4]]
F11.mat = measures[[5]]
F12.mat = measures[[6]]
accuracy.mat = measures[[7]]
one.error.mat = measures[[8]]
coverage.mat = measures[[9]]
rank.loss.mat = measures[[10]]
aveprecision.mat = measures[[11]]
```

```
      results.full[jjj,] =
            c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
            y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

#########################################################################
# Step 5:   Perform XGBoost on the relevant (k) features                #
#           k is the number of relevant features                        #
#########################################################################

      irrelevant.feat = which(apply(relmat,1,sum)==0)
      k = p - length(irrelevant.feat)
      irrelevant.feat.mat[jjj,1:length(irrelevant.feat)] =
            as.numeric(irrelevant.feat)

#########################################################################
# Remove irrelevant features from test and train split                  #
#########################################################################

      xydatatraink = as.matrix(xydatatrain[,-irrelevant.feat])
      xdatatraink = xydatatraink[,1:k]
      xydatatestk = as.matrix(xydatatest[,-irrelevant.feat])
      xdatatestk = xydatatestk[,1:k]
      Hloss.mat = matrix(0,nrow = 1, ncol = 1)
      clacc.mat = matrix(0, nrow = 1, ncol = 1)
      prec.mat = matrix(0, nrow = 1, ncol = 1)
      recall.mat = matrix(0, nrow = 1, ncol = 1)
      F11.mat = matrix(0, nrow = 1, ncol = 1)
      F12.mat = matrix(0, nrow = 1, ncol = 1)
      accuracy.mat = matrix(0, nrow = 1, ncol = 1)
      one.error.mat = matrix(0, nrow = 1, ncol = 1)
      coverage.mat = matrix(0, nrow = 1, ncol = 1)
      rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
      aveprecision.mat = matrix(0, nrow = 1, ncol = 1)
      Ntrain = nrow(xydatatraink)
      Ntest = nrow(xydatatestk)
      postprob = matrix(0, nrow = Ntest, ncol = q)
      Zmatind = matrix(0, nrow = Ntest, ncol = q)
      Zmatranks = matrix(0, nrow = Ntest, ncol = q)
```

```
      for (m in 1:q)    {
            model_xgb = xgboost(data = as.matrix(xdatatraink), label =
                  ydatatrain[,m], nrounds = ntreesvec[jjj,m], objective =
                  "binary:logistic", eta = eta[jjj,m], depth = depth[jjj,m],
                  verbose = 0)
            postprob[,m] = predict(model_xgb, newdata =
                  as.matrix(xdatatestk))
      }
      Zmatind[postprob>0.5]=1
      for (i in 1:Ntest) {
            Zmatranks[i,] = rank(-postprob[i,])
      }
      measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
      Hloss.mat = measures[[1]]
      clacc.mat = measures[[2]]
      prec.mat = measures[[3]]
      recall.mat = measures[[4]]
      F11.mat = measures[[5]]
      F12.mat = measures[[6]]
      accuracy.mat = measures[[7]]
      one.error.mat = measures[[8]]
      coverage.mat = measures[[9]]
      rank.loss.mat = measures[[10]]
      aveprecision.mat = measures[[11]]
      results.k[jjj,] =
            c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
            y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

#########################################################################
# Step 6:   Perform XGBoost:                                            #
#           Method SelectRank1: rank all features in group and          #
#           select feature with highest rank.                           #
#########################################################################

      irrelevant.feat = which(apply(relmat,1,sum)==0)
      k = p - length(irrelevant.feat)
      vec.index = as.vector(t(groupmatout))
```

```
      vec.index <- vec.index[vec.index != "0"]
      groupsize = rowSums(groupmatout!=0)
      Wvalues2 = Wvalues[vec.index,]
      cumsumgroupsize = cumsum(groupsize)
      averagerank.mat = matrix(0, nrow = p, ncol = q)
      countrow = 1
      for (countgroups in 1:length(groupsize)) {
            if (length(countrow:cumsumgroupsize[countgroups]) == 1)
                  averagerank.mat[countrow,] =  matrix(1,nrow = 1,ncol = q)
            else averagerank.mat[countrow:cumsumgroupsize[countgroups],] =
                  apply(-
                  Wvalues2[countrow:cumsumgroupsize[countgroups],],2,rank)
            countrow = cumsumgroupsize[countgroups]+1
      }
      averagerank = matrix(apply(averagerank.mat,1,mean),nrow = p)
      best = NULL
            for (g in 1:numberofgroups)   {
                  if (g==1) {begin=1
                  eindig = cumsumgroupsize[1]}
                        if (g>1) {begin=cumsumgroupsize[g-1]+1
                              eindig = cumsumgroupsize[g]}
                        index = which.min(averagerank[begin:eindig])
                        best[g]= groupmatout[g,index]
                  }
      last.out = intersect(best, as.vector(irrelevant.feat))
      best = best[best != last.out]
      best.mat[jjj,1:length(best)] =  best

#########################################################################
# Remove features from test and train split                             #
#########################################################################

      xdatatrainbest = xdatatrain[,best]
      xdatatestbest = xdatatest[,best]
      newp = length(best)
      Hloss.mat = matrix(0,nrow = 1, ncol = 1)
      clacc.mat = matrix(0, nrow = 1, ncol = 1)
      prec.mat = matrix(0, nrow = 1, ncol = 1)
```

```
recall.mat = matrix(0, nrow = 1, ncol = 1)
F11.mat = matrix(0, nrow = 1, ncol = 1)
F12.mat = matrix(0, nrow = 1, ncol = 1)
accuracy.mat = matrix(0, nrow = 1, ncol = 1)
one.error.mat = matrix(0, nrow = 1, ncol = 1)
coverage.mat = matrix(0, nrow = 1, ncol = 1)
rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
aveprecision.mat = matrix(0, nrow = 1, ncol = 1)
Ntrain = nrow(xydatatrain)
Ntest = nrow(xydatatest)
postprob = matrix(0, nrow = Ntest, ncol = q)
Zmatind = matrix(0, nrow = Ntest, ncol = q)
Zmatranks = matrix(0, nrow = Ntest, ncol = q)
for (m in 1:q) {
      model_xgb = xgboost(data = as.matrix(xdatatrainbest), label =
            ydatatrain[,m], nrounds = ntreesvec[jjj,m], objective =
            "binary:logistic", eta = eta[jjj,m], depth = depth[jjj,m],
            verbose = 0)
      postprob[,m] = predict(model_xgb, newdata =
            as.matrix(xdatatestbest))
}
Zmatind[postprob>0.5]=1
for (i in 1:Ntest) {
      Zmatranks[i,] = rank(-postprob[i,])
}
measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
Hloss.mat = measures[[1]]
clacc.mat = measures[[2]]
prec.mat = measures[[3]]
recall.mat = measures[[4]]
F11.mat = measures[[5]]
F12.mat = measures[[6]]
accuracy.mat = measures[[7]]
one.error.mat = measures[[8]]
coverage.mat = measures[[9]]
rank.loss.mat = measures[[10]]
aveprecision.mat = measures[[11]]
```

```
      results.best[jjj,] =
            c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
            y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

#########################################################################
# Step 7:   Perform XGBoost:                                            #
#           Method SelectRank2: rank all feautures in group and select  #
#           two features with highest ranks                             #
#           (if groupsize = 1, only one feature is included)            #
#########################################################################

      irrelevant.feat = which(apply(relmat,1,sum)==0)
      k = p - length(irrelevant.feat)
      vec.index = as.vector(t(groupmatout))
      vec.index <- vec.index[vec.index != "0"]
      groupsize = rowSums(groupmatout!=0)
      Wvalues2 = Wvalues[vec.index,]
      cumsumgroupsize = cumsum(groupsize)
      averagerank.mat = matrix(0, nrow = p, ncol = q)
      countrow = 1
      for (countgroups in 1:length(groupsize)) {
            if (length(countrow:cumsumgroupsize[countgroups]) == 1)
                  averagerank.mat[countrow,] =  matrix(1,nrow = 1,ncol = q)
            else averagerank.mat[countrow:cumsumgroupsize[countgroups],] =
                  apply(-
                  Wvalues2[countrow:cumsumgroupsize[countgroups],],2,rank)
      countrow = cumsumgroupsize[countgroups]+1
      }
      averagerank = matrix(apply(averagerank.mat,1,mean),nrow = p)
      best2 = matrix(0,nrow=numberofgroups,ncol=2)
      for (g in 1:numberofgroups)   {
            if (groupsize[g]==1) {best2[g,]=groupmatout[g,1]}
            if (groupsize[g]>1) {
                  if (g==1) {begin=1
                        eindig = cumsumgroupsize[1]}
                  if (g>1) {begin=cumsumgroupsize[g-1]+1
                        eindig = cumsumgroupsize[g]}
```

```
                x = sort(averagerank[begin:eindig],decreasing = FALSE,
                      index.return = TRUE)
                best2[g,]= groupmatout[g,x$ix[1:2]]
          }
    }
    best2 = as.vector(best2)
    best2 = unique(best2)
    last.out2 = intersect(best2, as.vector(irrelevant.feat))
    best2 = best2[best2 != last.out2[1]]
    best2 = best2[best2 != last.out2[2]]
    best2.mat[jjj,1:length(best2)] = best2

#########################################################################
# Remove features from test and train split.                            #
#########################################################################

    xdatatrainbest2 = xdatatrain[,best2]
    xdatatestbest2 = xdatatest[,best2]
    newp2 = length(best2)
    Hloss.mat = matrix(0,nrow = 1, ncol = 1)
    clacc.mat = matrix(0, nrow = 1, ncol = 1)
    prec.mat = matrix(0, nrow = 1, ncol = 1)
    recall.mat = matrix(0, nrow = 1, ncol = 1)
    F11.mat = matrix(0, nrow = 1, ncol = 1)
    F12.mat = matrix(0, nrow = 1, ncol = 1)
    accuracy.mat = matrix(0, nrow = 1, ncol = 1)
    one.error.mat = matrix(0, nrow = 1, ncol = 1)
    coverage.mat = matrix(0, nrow = 1, ncol = 1)
    rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
    aveprecision.mat = matrix(0, nrow = 1, ncol = 1)
    Ntrain = nrow(xydatatrain)
    Ntest = nrow(xydatatest)
    postprob = matrix(0, nrow = Ntest, ncol = q)
    Zmatind = matrix(0, nrow = Ntest, ncol = q)
    Zmatranks = matrix(0, nrow = Ntest, ncol = q)
    for (m in 1:q) {
          model_xgb = xgboost(data = as.matrix(xdatatrainbest2), label =
                ydatatrain[,m], nrounds = ntreesvec[jjj,m], objective =
```

```
            "binary:logistic", eta = eta[jjj,m], depth = depth[jjj,m],
            verbose = 0)
      postprob[,m] = predict(model_xgb, newdata =
            as.matrix(xdatatestbest2))
}
Zmatind[postprob>0.5]=1
      for (i in 1:Ntest)      {
            Zmatranks[i,] = rank(-postprob[i,])
}
measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
Hloss.mat = measures[[1]]
clacc.mat = measures[[2]]
prec.mat = measures[[3]]
recall.mat = measures[[4]]
F11.mat = measures[[5]]
F12.mat = measures[[6]]
accuracy.mat = measures[[7]]
one.error.mat = measures[[8]]
coverage.mat = measures[[9]]
rank.loss.mat = measures[[10]]
aveprecision.mat = measures[[11]]
results.best2[jjj,] =
      c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
      y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)
write.table(numberofgroups.mat, "C:\\...\\numberofgroups.mat.txt")
write.table(results.full, "C:\\...\\results.full.txt")
write.table(results.k, "C:\\...\\results.k.txt")
write.table(results.best, "C:\\...\\results.best.txt")
write.table(results.best2, "C:\\...\\results.best2.txt")

aver.numberofgroups = mean(numberofgroups.mat)
aver.results.full = apply(results.full,2,mean)
aver.results.k = apply(results.k,2,mean)
aver.results.best = apply(results.best,2,mean)
aver.results.best2 = apply(results.best2,2,mean)
med.results.full = apply(results.full,2,median)
med.results.k = apply(results.k,2,median)
med.results.best = apply(results.best,2,median)
```

```
      med.results.best2 = apply(results.best2,2,median)
      write.table(aver.numberofgroups, "C:\\...\\aver.numberofgroups.txt")
      write.table(aver.results.full, "C:\\...\\aver.results.full.txt")
      write.table(aver.results.k, "C:\\...\\aver.results.k.txt")
      write.table(aver.results.best, "C:\\...\\aver.results.best.txt")
      write.table(aver.results.best2, "C:\\...\\aver.results.best2.txt")
      write.table(med.results.full, "C:\\...\\med.results.full.txt")
      write.table(med.results.k, "C:\\...\\med.results.k.txt")
      write.table(med.results.best, "C:\\...\\med.results.best.txt")
      write.table(med.results.best2, "C:\\...\\med.results.best2.txt")
      write.table(irrelevant.feat.mat, "C:\\...\\irrelevant.feat.mat.txt")
      write.table(best.mat, "C:\\...\\best.mat.txt")
      write.table(best2.mat, "C:\\...\\best2.mat.txt")
      }
}
```

K.3 Generating synthetic multi-label datasets

```
function (N,k,pnoise,q,rho,pvek,Amat,signal)
{
#########################################################################
# N = number of instances                                               #
# k = number of relevant features                                       #
# pnoise = number of irrelevant features                                #
# q = number of labels                                                  #
# rho = correlations                                                    #
# pvek = label densities                                                #
# Amat = matrix used to specify local relevance of features for labels  #
# signal = strength of signal                                           #
#########################################################################

Amat = matrix(Amat,nrow = k)
n = length(pvek)
theta = qnorm(pvek)
sigmax = matrix(0.5,k,k)
diag(sigmax) = 1
apvek1 = NULL
xmat = matrix(0,N,(k+pnoise))
ymat = matrix(0,N,q)
```

```
apmat = matrix(0,q,q)
gem = rep(0,k)
sign.rho = sign(rho)
for (j in 1:k) apvek1[j] = sum(Amat[j,]*pvek)
      for (k1 in 1:q) for (k2 in 1:q) {
            term1 = rho*sqrt(pvek[k1]*(1-pvek[k1])*pvek[k2]*(1-pvek[k2]))
            term2 = pvek[k1]*pvek[k2]
            term3 = t(Amat[,k1])%*%Amat[,k2]
            apmat[k1,k2] = term3*(term1+term2)
      }
      antwoord = sum(apvek1)+sum(apmat)-sum(diag(apmat))-(sum(apvek1^2))
c = sqrt(signal/antwoord)
itel = 0
while (itel < N) {
      if (rho >= 0) {
            eps0 = rnorm(1)
            eps = rnorm(n)
            u = rbinom(n,1,sqrt(rho))
            z = u*eps0+(1-u)*eps
            yvek = (z <= theta)+0
      }
      if (rho<0) {
            eps = rnorm(n)
            z = eps
            u = rbinom(n,1,abs(rho))
            for (j in 2:n) {
                  z[j] = sign.rho*u[j]*z[j-1]+(1-u[j])*eps[j]
            }
            yvek = (z <= theta)+0
      }
      if (sum(yvek) >= 0) {
            itel = itel+1
            for (j in 1:k) gem[j] = c*sum(Amat[j,]*yvek)
            xmat[itel,] =
                  c(mvrnorm(1,gem,sigmax),mvrnorm(1,rep(0,pnoise),diag(pnoise
                  )))
            ymat[itel,] = yvek
      }
```

```
}
return(list(xmat,ymat,antwoord,c))
}
```

K.4 Implementation of RPFS procedure based on IG using SVM classifier for synthetic datasets

```
function (cutoff,nrep)
{
#########################################################################
# For Synthetic data set                                                #
# p refers to the number of features                                    #
# q refers to the number of labels                                      #
# k refers to the number of relevant features                           #
# N refers to the number of instances                                   #
#     cutoff = is a specified threshold                                 #
#     if the IG is >= cutoff then the feature is deemed relevant (1)    #
#     if the IG is < cutoff then the feature is deemed irrelevant (0)   #
#########################################################################

#########################################################################
# Step 1:   Load libraries and generate data                            #
#########################################################################

library(MASS)
library(miscFuncs)
library(CORElearn)
library(UBbipl)
library(UBFigs)
library(ggplot2)
```

```
#########################################################################
# Initialise the following:                                             #
# 1. numberofgroups.mat = nrep x 1 matrix that captures the number      #
#     of groups used for each repetition                                #
# 2. results.full, results.k, results.best and results.best2 =          #
#     nrep x 11 matrices that contains the evaluation measures for      #
#     each repetition                                                   #
# 3. irrelevant.feat.mat = nrep x p matrix that captures the            #
#     irrelevant features for each repetition                           #
# 4. best.mat = nrep x p matrix that captures the feature with the      #
#     highest rank in each group                                        #
# 5. best2.mat = nrep x p matrix that captures the two features with    #
#     the two highest ranks in each group                               #
#########################################################################

numberofgroups.mat = matrix(0, nrow = nrep, ncol = 1)
results.full = matrix(0,nrow = nrep, ncol = 11)
results.k = matrix(0,nrow = nrep, ncol = 11)
results.best = matrix(0,nrow = nrep, ncol = 11)
results.best2 = matrix(0,nrow = nrep, ncol = 11)
irrelevant.feat.mat = matrix(0,nrow = nrep, ncol = p)
best.mat = matrix(0,nrow = nrep, ncol = p)
best2.mat = matrix(0,nrow = nrep, ncol = p)
numberofirrelevantfeat.mat = matrix(0,nrow = nrep, ncol = 1)

#########################################################################
# The MC loop now starts.                                               #
#########################################################################

for (jjj in 1:nrep) {
      print(jjj)
      synthdatatrain = synth(N = 80,k = 10,pnoise = 10,q = 6,rho = 0,pvek =
      c(0.4, 0.4, 0.4, 0.4, 0.4, 0.4), Amat = Amat, signal = 10)
      ydatatrain = synthdatatrain[[2]]
      xdatatrain = synthdatatrain[[1]]
      xydatatrain = cbind(xdatatrain,ydatatrain)
      synthdatatest = synth(N = 10000 ,k = 10,pnoise = 10,q = 6,rho = 0,pvek
      = c(0.4, 0.4, 0.4, 0.4, 0.4, 0.4), Amat = Amat, signal = 10)
```

```
      ydatatest = synthdatatest[[2]]
      xdatatest = synthdatatest[[1]]
      xydatatest = cbind(xdatatest,ydatatest)
      Nnew = nrow(xydatatest)
      p = ncol(xdatatrain)
      q = ncol(ydatatrain)

#########################################################################
# Step 2:   Find relevant features and set-up relevance matrix          #
#                 Information Gain                                      #
#########################################################################

      IGvalues = matrix(rep(0),ncol = q,nrow = p)
      q = ncol(ydatatrain)
      for (i in 1:q)    {
            yvec = as.factor(ydatatrain[,i])
            datatrain = data.frame(yvec,xdatatrain)
            value = attrEval(as.factor(yvec)~.,datatrain,estimator="InfGain")
            for (j in 1:p) IGvalues[j,i]=value[j]
      }
      IGrelmat = matrix(0,nrow=p,ncol=q)
      IGrelmat[IGvalues>cutoff]=1

      write.table(IGrelmat,"C:\\...\\SynthM3relmat.txt")
      write.table(IGvalues,"C:\\...\\SynthM3IGvalues.txt")

#########################################################################
# Output:                                                               #
# relmat = pxq relevance matrix                                         #
# IGvalue = IGvalues obtained from Information Gain                     #
#########################################################################

#########################################################################
# Step 3:   Create coordinatets for MCA biplot representing groups      #
#           Create groups based on MCA                                  #
#########################################################################

      relmat = read.table("C:\\...\\SynthM3relmat.txt")
```

```
      Z.02 = MCABIPL(relmat)$Z.0
      Z2 = MCABIPL(relmat)$Z
      reltemp2 = Z.02
      p = nrow(reltemp2)
      q = ncol(reltemp2)
      deletedrow = matrix(-1,nrow = 1, ncol = q)
      groupmat = matrix(0, nrow = p, ncol = p)
      uniquerows = matrix(0, nrow = p, ncol = q)
      ngroup = 0
      maxningroup = 0
      for (count1 in 1:p) {
            thisgroup = reltemp2[count1,]
            (sum(thisgroup == deletedrow) < q)  {
            ngroup = ngroup + 1
            uniquerows[ngroup,] = as.matrix(thisgroup)
            groupmat[ngroup,1] = count1
            ningroup = 1
            if (count1 < p)
                  for (count2 in (count1+1):p)
                        if (sum(reltemp2[count2,] == thisgroup) == q) {
                              ningroup = ningroup + 1
                              maxningroup = max(maxningroup,ningroup)
                              groupmat[ngroup, ningroup] = count2
                              reltemp2[count2,] = as.matrix(deletedrow)
                        }
            }
      }
      groupmatout = groupmat[1:ngroup,1:maxningroup]
      uniquerowsout = uniquerows[1:ngroup,]
      groupsize = rowSums(groupmatout!=0)
      numberofgroups = nrow(groupmatout)
      numberofgroups.mat[jjj,1] = numberofgroups

#########################################################################
# Step 4: Perform SVM on the full set of p features                     #
#########################################################################

      Hloss.mat = matrix(0,nrow = 1, ncol = 1)
```

```
      clacc.mat = matrix(0, nrow = 1, ncol = 1)
      prec.mat = matrix(0, nrow = 1, ncol = 1)
      recall.mat = matrix(0, nrow = 1, ncol = 1)
      F11.mat = matrix(0, nrow = 1, ncol = 1)
      F12.mat = matrix(0, nrow = 1, ncol = 1)
      accuracy.mat = matrix(0, nrow = 1, ncol = 1)
      one.error.mat = matrix(0, nrow = 1, ncol = 1)
      coverage.mat = matrix(0, nrow = 1, ncol = 1)
      rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
      aveprecision.mat = matrix(0, nrow = 1, ncol = 1)

#########################################################################
# Binary Relevance (BR)                                                 #
#########################################################################

      Ntrain = nrow(xydatatrain)
      Ntest = nrow(xydatatest)
      Fmat_svm = matrix(0, nrow = Ntest, ncol = q)
      Zmatind = matrix(0, nrow = Ntest, ncol = q)
      Zmatranks = matrix(0, nrow = Ntest, ncol = q)
      aver = matrix(apply(xdatatrain[,1:ncol(xdatatrain)],2,mean), ncol =
            ncol(xdatatrain))
      varian = matrix(apply(xdatatrain[,1:ncol(xdatatrain)],2,var), ncol =
            ncol(xdatatrain))
      xdatatrainscalesvm = scale(xdatatrain)
      xdatatestscalesvm = scale(xdatatest,center = aver,scale = sqrt(varian))
      rbf = rbfdot(sigma = 1/p)
      gmat = kernelMatrix(rbf,xdatatestscalesvm,xdatatrainscalesvm)
      for (j in 1:q) {
            ydatatrainsvm = ydatatrain[,j]
            ydatatrainsvm[ydatatrain[,j]==0] = -1
            ydatatrainsvm = factor(ydatatrainsvm)
            svmfit = ksvm(x=xdatatrainscalesvm,y=ydatatrainsvm,scaled =
                  TRUE,type = "C-svc", kernel = "rbfdot", kpar =
                  list(sigma=1/p), C = 1, prob.model = FALSE)
            coefsvm = as.matrix(unlist(coef(svmfit)))
            bcoef = unlist(b(svmfit))
            indeks = unlist(SVindex(svmfit))
```

```
            coefVol = rep(0,Ntrain)
            coefVol[indeks] = coefsvm
            q3 = matrix(t(as.matrix(coefVol))%*%t(gmat),ncol=1)
            fvalues = q3-bcoef[1]
            Fmat_svm[,j] = fvalues
      }
      for (i in 1:Ntest) {
            av = Fmat_svm[i,]
            avs = sort(av,decreasing=TRUE,index.return=TRUE)
            Zmatind[i,avs$ix[1:3]] = 1
            Zmatranks[i,] = rank(-av)
      }
      measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
      Hloss.mat = measures[[1]]
      clacc.mat = measures[[2]]
      prec.mat = measures[[3]]
      recall.mat = measures[[4]]
      F11.mat = measures[[5]]
      F12.mat = measures[[6]]
      accuracy.mat = measures[[7]]
      one.error.mat = measures[[8]]
      coverage.mat = measures[[9]]
      rank.loss.mat = measures[[10]]
      aveprecision.mat = measures[[11]]
      results.full[jjj,] =
            c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
            y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

#########################################################################
# Step 5:   Perform SVM on the relevant (k) features                    #
#                 k is the number of relevant features                  #
#########################################################################

      irrelevant.feat = which(apply(relmat,1,sum)==0)
      k = p - length(irrelevant.feat)
      numberofirrelevantfeat.mat[jjj,1] = length(irrelevant.feat)
      function (k)
            {if (k == 0){
```

```
                stop("No relevant features")
            }
        }

#########################################################################
# Remove irrelevant features from test and train split                  #
#########################################################################

        xydatatraink = as.matrix(xydatatrain[,-irrelevant.feat])
        xdatatraink = xydatatraink[,1:k]
        xydatatestk = as.matrix(xydatatest[,-irrelevant.feat])
        xdatatestk = xydatatestk[,1:k]
        Hloss.mat = matrix(0,nrow = 1, ncol = 1)
        clacc.mat = matrix(0, nrow = 1, ncol = 1)
        prec.mat = matrix(0, nrow = 1, ncol = 1)
        recall.mat = matrix(0, nrow = 1, ncol = 1)
        F11.mat = matrix(0, nrow = 1, ncol = 1)
        F12.mat = matrix(0, nrow = 1, ncol = 1)
        accuracy.mat = matrix(0, nrow = 1, ncol = 1)
        one.error.mat = matrix(0, nrow = 1, ncol = 1)
        coverage.mat = matrix(0, nrow = 1, ncol = 1)
        rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
        aveprecision.mat = matrix(0, nrow = 1, ncol = 1)

#########################################################################
# Binary Relevance (BR)                                                 #
#########################################################################

        Ntrain = nrow(xydatatraink)
        Ntest = nrow(xydatatestk)
        Fmat_svm = matrix(0, nrow = Ntest, ncol = q)
        Zmatind = matrix(0, nrow = Ntest, ncol = q)
        Zmatranks = matrix(0, nrow = Ntest, ncol = q)
        aver = matrix(apply(xdatatraink[,1:ncol(xdatatraink)],2,mean), ncol =
              ncol(xdatatraink))
        varian = matrix(apply(xdatatraink[,1:ncol(xdatatraink)],2,var), ncol =
              ncol(xdatatraink))
        xdatatrainscalesvm = scale(xdatatraink)
```

```
xdatatestscalesvm = scale(xdatatestk,center = aver,scale =
      sqrt(varian))
rbf = rbfdot(sigma = 1/k)
gmat = kernelMatrix(rbf,xdatatestscalesvm,xdatatrainscalesvm)
for (j in 1:q) {
      ydatatrainsvm = ydatatrain[,j]
      ydatatrainsvm[ydatatrain[,j]==0] = -1
      datatrainsvm = factor(ydatatrainsvm)
      svmfit = ksvm(x=xdatatrainscalesvm,y=ydatatrainsvm,scaled =
            TRUE,type = "C-svc", kernel = "rbfdot", kpar =
            list(sigma=1/k), C = 1, prob.model = FALSE)
      coefsvm = as.matrix(unlist(coef(svmfit)))
      bcoef = unlist(b(svmfit))
      indeks = unlist(SVindex(svmfit))
      coefVol = rep(0,Ntrain)
      coefVol[indeks] = coefsvm
      q3 = matrix(t(as.matrix(coefVol))%*%t(gmat),ncol=1)
      fvalues = q3-bcoef[1]
      Fmat_svm[,j] = fvalues
}
for (i in 1:Ntest) {
      av = Fmat_svm[i,]
      avs = sort(av,decreasing=TRUE,index.return=TRUE)
      Zmatind[i,avs$ix[1:3]] = 1
      Zmatranks[i,] = rank(-av)
}
measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
Hloss.mat = measures[[1]]
clacc.mat = measures[[2]]
prec.mat = measures[[3]]
recall.mat = measures[[4]]
F11.mat = measures[[5]]
F12.mat = measures[[6]]
accuracy.mat = measures[[7]]
one.error.mat = measures[[8]]
coverage.mat = measures[[9]]
rank.loss.mat = measures[[10]]
aveprecision.mat = measures[[11]]
```

```
      results.k[jjj,] =
            c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
            y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

#########################################################################
# Step 6:   Perform SVM:                                                #
#           Method SelectRank1: rank all features in group and select   #
#           feature with highest rank                                   #
#########################################################################

      irrelevant.feat = which(apply(relmat,1,sum)==0)
      k = p - length(irrelevant.feat)
      function (k)
      {if (k == 0){
            stop("No relevant features")
            }
      }
      vec.index = as.vector(t(groupmatout))
      vec.index <- vec.index[vec.index != "0"]
      groupsize = rowSums(groupmatout!=0)
      Wvalues2 = IGvalues[vec.index,]
      cumsumgroupsize = cumsum(groupsize)
      averagerank.mat = matrix(0, nrow = p, ncol = q)
      countrow = 1
      for (countgroups in 1:length(groupsize)) {
            if (length(countrow:cumsumgroupsize[countgroups]) == 1)
                  averagerank.mat[countrow,] =  matrix(1,nrow = 1,ncol = q)
            else averagerank.mat[countrow:cumsumgroupsize[countgroups],] =
                  apply(-
                  Wvalues2[countrow:cumsumgroupsize[countgroups],],2,rank)
            countrow = cumsumgroupsize[countgroups]+1
      }
      averagerank = matrix(apply(averagerank.mat,1,mean),nrow = p)
      best = NULL
      for (g in 1:numberofgroups)   {
            if (g==1) {begin=1
                  eindig = cumsumgroupsize[1]}
            if (g>1) {begin=cumsumgroupsize[g-1]+1
```

```
            eindig = cumsumgroupsize[g]}
            index = which.min(averagerank[begin:eindig])
            best[g]= groupmatout[g,index]
      }
      last.out = intersect(best, as.vector(irrelevant.feat))
      best = best[best != last.out]

#############################################################################
# Remove features from test and train split                                 #
#############################################################################

      xdatatrainbest = xdatatrain[,best]
      xdatatestbest = xdatatest[,best]
      newp = length(best)
      Hloss.mat = matrix(0,nrow = 1, ncol = 1)
      clacc.mat = matrix(0, nrow = 1, ncol = 1)
      prec.mat = matrix(0, nrow = 1, ncol = 1)
      recall.mat = matrix(0, nrow = 1, ncol = 1)
      F11.mat = matrix(0, nrow = 1, ncol = 1)
      F12.mat = matrix(0, nrow = 1, ncol = 1)
      accuracy.mat = matrix(0, nrow = 1, ncol = 1)
      one.error.mat = matrix(0, nrow = 1, ncol = 1)
      coverage.mat = matrix(0, nrow = 1, ncol = 1)
      rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
      aveprecision.mat = matrix(0, nrow = 1, ncol = 1)

#############################################################################
# Binary Relevance (BR)                                                     #
#############################################################################

      Ntrain = nrow(xydatatrain)
      Ntest = nrow(xydatatest)
      Fmat_svm = matrix(0, nrow = Ntest, ncol = q)
      Zmatind = matrix(0, nrow = Ntest, ncol = q)
      Zmatranks = matrix(0, nrow = Ntest, ncol = q)
      aver = matrix(apply(xdatatrainbest[,1:ncol(xdatatrainbest)],2,mean),
            ncol = ncol(xdatatrainbest))
```

```
varian = matrix(apply(xdatatrainbest[,1:ncol(xdatatrainbest)],2,var),
      ncol = ncol(xdatatrainbest))
xdatatrainscalesvm = scale(xdatatrainbest)
xdatatestscalesvm = scale(xdatatestbest,center = aver,scale =
      sqrt(varian))
rbf = rbfdot(sigma = 1/newp)
gmat = kernelMatrix(rbf,xdatatestscalesvm,xdatatrainscalesvm)
for (j in 1:q) {
      ydatatrainsvm = ydatatrain[,j]
      ydatatrainsvm[ydatatrain[,j]==0] = -1
      ydatatrainsvm = factor(ydatatrainsvm)
      svmfit = ksvm(x=xdatatrainscalesvm,y=ydatatrainsvm,scaled =
            TRUE,type = "C-svc", kernel = "rbfdot", kpar =
            list(sigma=1/newp), C = 1, prob.model = FALSE)
      coefsvm = as.matrix(unlist(coef(svmfit)))
      bcoef = unlist(b(svmfit))
      indeks = unlist(SVindex(svmfit))
      coefVol = rep(0,Ntrain)
      coefVol[indeks] = coefsvm
      q3 = matrix(t(as.matrix(coefVol))%*%t(gmat),ncol=1)
      fvalues = q3-bcoef[1]
      Fmat_svm[,j] = fvalues
}
for (i in 1:Ntest) {
      av = Fmat_svm[i,]
      avs = sort(av,decreasing=TRUE,index.return=TRUE)
      Zmatind[i,avs$ix[1:3]] = 1
      Zmatranks[i,] = rank(-av)
}
measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
Hloss.mat = measures[[1]]
clacc.mat = measures[[2]]
prec.mat = measures[[3]]
recall.mat = measures[[4]]
F11.mat = measures[[5]]
F12.mat = measures[[6]]
accuracy.mat = measures[[7]]
one.error.mat = measures[[8]]
```

```
      coverage.mat = measures[[9]]
      rank.loss.mat = measures[[10]]
      aveprecision.mat = measures[[11]]
      results.best[jjj,] =
            c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
            y.mat,one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

#########################################################################
# Step 7:   Perform SVM:                                                #
#           Method SelectRank2: rank all feautures in group and select  #
#           two features with highest ranks                             #
#           (if groupsize = 1, only one feature is included)            #
#########################################################################

      irrelevant.feat = which(apply(relmat,1,sum)==0)
      k = p - length(irrelevant.feat)
      vec.index = as.vector(t(groupmatout))
      vec.index <- vec.index[vec.index != "0"]
      groupsize = rowSums(groupmatout!=0)
      Wvalues2 = IGvalues[vec.index,]
      cumsumgroupsize = cumsum(groupsize)
      averagerank.mat = matrix(0, nrow = p, ncol = q)
      countrow = 1
      for (countgroups in 1:length(groupsize)) {
            if (length(countrow:cumsumgroupsize[countgroups]) == 1)
                  averagerank.mat[countrow,] =  matrix(1,nrow = 1,ncol = q)
            else averagerank.mat[countrow:cumsumgroupsize[countgroups],] =
                  apply(-
                  Wvalues2[countrow:cumsumgroupsize[countgroups],],2,rank)
            countrow = cumsumgroupsize[countgroups]+1
      }
      averagerank = matrix(apply(averagerank.mat,1,mean),nrow = p)
      best2 = matrix(0,nrow=numberofgroups,ncol=2)
      for (g in 1:numberofgroups)   {
            if (groupsize[g]==1) {best2[g,]=groupmatout[g,1]}
            if (groupsize[g]>1) {
                  if (g==1) {begin=1
                  eindig = cumsumgroupsize[1]}
```

```
                if (g>1) {begin=cumsumgroupsize[g-1]+1
                eindig = cumsumgroupsize[g]}
                x = sort(averagerank[begin:eindig],decreasing = FALSE,
                      index.return = TRUE)
               best2[g,]= groupmatout[g,x$ix[1:2]]
            }
      }
      best2 = as.vector(best2)
      best2 = unique(best2)
      last.out2 = intersect(best2, as.vector(irrelevant.feat))
      best2 = best2[best2 != last.out2[1]]
      best2 = best2[best2 != last.out2[2]]

#############################################################################
# Remove features from test and train split                                 #
#############################################################################

      xdatatrainbest2 = xdatatrain[,best2]
      xdatatestbest2 = xdatatest[,best2]
      newp2 = length(best2)
      Hloss.mat = matrix(0,nrow = 1, ncol = 1)
      clacc.mat = matrix(0, nrow = 1, ncol = 1)
      prec.mat = matrix(0, nrow = 1, ncol = 1)
      recall.mat = matrix(0, nrow = 1, ncol = 1)
      F11.mat = matrix(0, nrow = 1, ncol = 1)
      F12.mat = matrix(0, nrow = 1, ncol = 1)
      accuracy.mat = matrix(0, nrow = 1, ncol = 1)
      one.error.mat = matrix(0, nrow = 1, ncol = 1)
      coverage.mat = matrix(0, nrow = 1, ncol = 1)
      rank.loss.mat = matrix(0, nrow = 1, ncol = 1)
      aveprecision.mat = matrix(0, nrow = 1, ncol = 1)

#############################################################################
# Binary Relevance (BR)                                                     #
#############################################################################

      Ntrain = nrow(xydatatrain)
      Ntest = nrow(xydatatest)
```

```
Fmat_svm = matrix(0, nrow = Ntest, ncol = q)
Zmatind = matrix(0, nrow = Ntest, ncol = q)
Zmatranks = matrix(0, nrow = Ntest, ncol = q)
aver = matrix(apply(xdatatrainbest2[,1:ncol(xdatatrainbest2)],2,mean),
      ncol = ncol(xdatatrainbest2))
varian = matrix(apply(xdatatrainbest2[,1:ncol(xdatatrainbest2)],2,var),
      ncol = ncol(xdatatrainbest2))
xdatatrainscalesvm = scale(xdatatrainbest2)
xdatatestscalesvm = scale(xdatatestbest2,center = aver,scale =
      sqrt(varian))
rbf = rbfdot(sigma = 1/newp2)
gmat = kernelMatrix(rbf,xdatatestscalesvm,xdatatrainscalesvm)
for (j in 1:q) {
      ydatatrainsvm = ydatatrain[,j]
      ydatatrainsvm[ydatatrain[,j]==0] = -1
      ydatatrainsvm = factor(ydatatrainsvm)
            svmfit = ksvm(x=xdatatrainscalesvm,y=ydatatrainsvm,scaled =
                  TRUE,type = "C-svc", kernel = "rbfdot", kpar =
                  list(sigma=1/newp2), C = 1, prob.model = FALSE)
      coefsvm = as.matrix(unlist(coef(svmfit)))
      bcoef = unlist(b(svmfit))
      indeks = unlist(SVindex(svmfit))
      coefVol = rep(0,Ntrain)
      coefVol[indeks] = coefsvm
      q3 = matrix(t(as.matrix(coefVol))%*%t(gmat),ncol=1)
      fvalues = q3-bcoef[1]
      Fmat_svm[,j] = fvalues
}
for (i in 1:Ntest) {
      av = Fmat_svm[i,]
      avs = sort(av,decreasing=TRUE,index.return=TRUE)
      Zmatind[i,avs$ix[1:3]] = 1
      Zmatranks[i,] = rank(-av)
      }
measures=Fmeasures(ydatatest,Zmatind,Zmatranks)
Hloss.mat = measures[[1]]
clacc.mat = measures[[2]]
prec.mat = measures[[3]]
```

```
recall.mat = measures[[4]]
F11.mat = measures[[5]]
F12.mat = measures[[6]]
accuracy.mat = measures[[7]]
one.error.mat = measures[[8]]
coverage.mat = measures[[9]]
rank.loss.mat = measures[[10]]
aveprecision.mat = measures[[11]]
results.best2[jjj,] =
      c(Hloss.mat,clacc.mat,prec.mat,recall.mat,F11.mat,F12.mat,accurac
      y.mat, one.error.mat,coverage.mat,rank.loss.mat,aveprecision.mat)

write.table(numberofgroups.mat, "C:\\...\\numberofgroups.mat.txt")
write.table(results.full, "C:\\...\\results.full.txt")
write.table(results.k, "C:\\...\\results.k.txt")
write.table(results.best, "C:\\...\\results.best.txt")
write.table(results.best2, "C:\\...\\results.best2.txt")

aver.numberofgroups = mean(numberofgroups.mat)
aver.numberofirrelevantfeatures = mean(numberofirrelevantfeat.mat)
aver.results.full = apply(results.full,2,mean)
aver.results.k = apply(results.k,2,mean)
aver.results.best = apply(results.best,2,mean)
aver.results.best2 = apply(results.best2,2,mean)
med.results.full = apply(results.full,2,median)
med.results.k = apply(results.k,2,median)
med.results.best = apply(results.best,2,median)
med.results.best2 = apply(results.best2,2,median)

write.table(aver.numberofgroups, "C:\\...\\aver.numberofgroups.txt")
write.table(aver.numberofirrelevantfeatures,
      "C:\\...\\aver.numberofirrelevantfeatures.txt")
write.table(aver.results.full, "C:\\...\\aver.results.full.txt")
write.table(aver.results.k, "C:\\...\\aver.results.k.txt")
write.table(aver.results.best, "C:\\...\\aver.results.best.txt")
write.table(aver.results.best2, "C:\\...\\aver.results.best2.txt")
write.table(med.results.full, "C:\\...\\med.results.full.txt")
write.table(med.results.k, "C:\\...\\med.results.k.txt")
```

```
      write.table(med.results.best, "C:\\...\\med.results.best.txt")
      write.table(med.results.best2, "C:\\...\\med.results.best2.txt")
      }
}
```

K.5 Implementation of FS procedures proposed by Sandrock and Steel (2016) and Spolaôr *et al*. (2013) on synthetic datasets

```
function (p,alpha,Bselect,K,M)
{
#############################################################################
# alpha, Bselect are parameters for the probe selection procedure           #
# K denotes the number of labels                                            #
# trainfrac is the fraction of the benchmark dataset to be used             #
#     for training - this will be redundant if synthetic data are used      #
# M is the number of times that the dataset has to be split into            #
#     training and testing sets.                                            #
#############################################################################

#############################################################################
# Call the library packages required                                        #
#############################################################################

library(e1071)
library(kernlab)
library(foreign)
library(COREElearn)

#############################################################################
# We now have a number of sub-functions, before we get to the main          #
# program                                                                   #
#############################################################################

threshold = function(fixedt,ydata,ftraindata,ftestdata)
{
```

```
#########################################################################
# This program takes 3 matrices as input:                               #
# 1. A matrix of training data labels.                                  #
# 2. A matrix of f-values to threshold.                                 #
# It then employs different thresholding approaches to transform the    #
#      f-values to labels.                                              #
#########################################################################

ymat = as.matrix(ydata)
ftrainmat = as.matrix(ftraindata)
ftestmat = as.matrix(ftestdata)
Ntrain = nrow(ymat)
Ntest = nrow(ftestmat)
Kval = ncol(ftestmat)
ylabels1 = matrix(0,Ntest,Kval)
ylabelsrank1 = matrix(0,Ntest,Kval)
ylabels2 = matrix(0,Ntest,Kval)
ylabelsrank2 = matrix(0,Ntest,Kval)

#########################################################################
# Fixed threshold procedure                                             #
#########################################################################

      for (i in 1:Ntest) {
            ylabels1[i,] = as.numeric(ftestdata[i,]>fixedt)
            vector = ftestdata[i,]
            indices = sort(vector,decreasing=TRUE,index.return=TRUE)$ix
            for (k in 1:Kval) ylabelsrank1[i,indices[k]] = k
      }

#########################################################################
# Quantile threshold procedure                                          #
#########################################################################

threshold2 = rep(0,Kval)
densities = apply(ymat,2,mean)
for (k in 1:Kval) {
      vector = ftestdata[,k]
```

```
      threshold2[k] = quantile(vector,1-densities[k],names=FALSE)
      for (i in 1:Ntest) {
            ylabels2[i,k] = as.numeric(ftestdata[i,k]>threshold2[k])
            ylabelsrank2[i,k] = ylabelsrank1[i,k]
            }
      }

#########################################################################
# Return the output.                                                    #
# Currently it is only the output from the quantile threshold approach  #
#     that is returned.                                                 #
#########################################################################

Output = list(ylabels2,ylabelsrank2)
return(Output)
}

#########################################################################
# This function computes the different measures for evaluating the      #
#     performance of a ML approach.                                     #
#########################################################################

measures = function(ylabels,zlabels,rankedlabels) {

Nnew = nrow(ylabels)
q = ncol(ylabels)
yminz = ylabels-zlabels
yprodz = ylabels*zlabels
ydeltaz = apply(yminz,1,function(x) sum(abs(x)))
nylabels = apply(ylabels,1,sum)
nzlabels = apply(zlabels,1,sum)
somnylablesnzlabels = (nylabels + nzlabels)
proportion1 = ydeltaz/q
yintersectionz = apply(yprodz,1,sum)
yunionz = nylabels + nzlabels - yintersectionz
proportion2 = yintersectionz[nzlabels>0]/nzlabels[nzlabels>0]
proportion3 = yintersectionz[nylabels>0]/nylabels[nylabels>0]
```

```
proportion4 =
      yintersectionz[somnylablesnzlabels>0]/somnylablesnzlabels[somnylablesnz
      labels>0]
proportion5 = yintersectionz[nylabels>0]/yunionz[nylabels>0]
Hloss = mean(proportion1)
precision = mean(proportion2)
recall = mean(proportion3)
accuracy = mean(proportion5)
ylabtimesrank = ylabels*rankedlabels
not.ylabtimesrank = (matrix(1,nrow=Nnew,ncol=q)-ylabels)*rankedlabels
one.error = mean(apply(ylabtimesrank,1,function(x) min(x[x>0])!=1))
coverage =
      mean(apply(ylabtimesrank[nylabels>0,],1,max)/apply(ylabels[nylabels>0,]
      ,1,sum))-1
output = list(Hloss,precision,recall,accuracy,one.error,coverage)
return(output)
}

#########################################################################
# This function performs probe variable selection.                      #
#########################################################################

selectionfour = function(alfa, B, K, xmat, ymat) {

alfaB = floor(alfa * B)
xmat = as.matrix(xmat)
ymat = as.matrix(ymat)
N = nrow(xmat)
p = ncol(xmat)
AAmat=array(0,c(K,p,3))
Zmat = matrix(0, N, p)
Lmat = ymat
CormatLX = matrix(0, K, p)
CormatLZ = rep(0, K * p * B)
dim(CormatLZ) = c(K, p, B)
```

```
#########################################################################
# Use the correlation coefficient to quantify the importance of an      #
#     input for the response.                                           #
#########################################################################

for (k in 1:K) for (j in 1:p) {
      yvek=Lmat[,k]
      xvek=xmat[,j]
      CormatLX[k, j] = fimpcor(yvek,xvek)
}
for (ir in 1:B) {
      indekse = sample(1:N, N, replace = FALSE)
      Zmat = xmat[indekse, ]
      for (k in 1:K) for (j in 1:p) {
            yvek=Lmat[,k]
            xvek=Zmat[,j]
            CormatLZ[k, j, ir] = fimpcor(yvek,xvek)
      }
}
Cmat = matrix(0, K, p)
Amat = matrix(0, K, p)
for (k in 1:K) for (j in 1:p) {
      rvec = CormatLZ[k, j, ]
      wvec = sort(rvec, decreasing = FALSE)
      Cmat[k, j] = wvec[alfaB]
}
for (k in 1:K) for (j in 1:p) {
      if (CormatLX[k, j] > Cmat[k,j])
            Amat[k, j] = 1
}
AAmat[,,1]=Amat
```

```
#########################################################################
# Use ReliefF to quantify the importance of an input for the response.  #
#########################################################################

for (k in 1:K) {
      yvek=Lmat[,k]
      CormatLX[k,] = fimprelf(yvek,xmat)
}
for (ir in 1:B) {
      indekse = sample(1:N, N, replace = FALSE)
      Zmat = xmat[indekse, ]
            for (k in 1:K) {
                  yvek=Lmat[,k]
                  CormatLZ[k,,ir] = fimprelf(yvek,Zmat)
            }
}
Cmat = matrix(0, K, p)
Amat = matrix(0, K, p)
for (k in 1:K) for (j in 1:p) {
      rvec = CormatLZ[k, j, ]
      wvec = sort(rvec, decreasing = FALSE)
      Cmat[k, j] = wvec[alfaB]
}
for (k in 1:K) for (j in 1:p) {
            if (CormatLX[k, j] > Cmat[k,j])
            Amat[k, j] = 1
}
AAmat[,,2]=Amat

#########################################################################
# Use information gain to quantify the importance of an input for the   #
#     response.                                                         #
#########################################################################

for (k in 1:K) {
      yvek=Lmat[,k]
      CormatLX[k,] = fimpinfg(yvek,xmat)
}
```

```
for (ir in 1:B) {
      indekse = sample(1:N, N, replace = FALSE)
      Zmat = xmat[indekse, ]
      for (k in 1:K) {
            yvek=Lmat[,k]
            CormatLZ[k,,ir] = fimpinfg(yvek,Zmat)
      }
}
Cmat = matrix(0, K, p)
Amat = matrix(0, K, p)
for (k in 1:K) for (j in 1:p) {
      rvec = CormatLZ[k, j, ]
      wvec = sort(rvec, decreasing = FALSE)
      Cmat[k, j] = wvec[alfaB]
}
for (k in 1:K) for (j in 1:p) {
      if (CormatLX[k, j] > Cmat[k,j])
      Amat[k, j] = 1
}
AAmat[,,3]=Amat
for (k in 1:K) {
      yvek=Lmat[,k]
      CormatLX[k,] = fimprelf(yvek,xmat)
}
output=list(AAmat,CormatLX)
return(output)
}

#########################################################################
# We now have the functions computing the different variable importance #
#     measures.                                                         #
#########################################################################

fimpcor = function(y,x)
{
abs(cor(y,x))
}
fimprelf = function(y,x)
```

```
{
Fwaardes=NULL
p=ncol(x)
mydata=data.frame(y,x)
waarde=attrEval(as.factor(y)~.,mydata,estimator="ReliefFequalK")
for (j in 1:p) Fwaardes[j]=waarde[j]
return(Fwaardes)
}
fimpinfg = function(y,x)
{
Fwaardes=NULL
p=ncol(x)
Ytrain=as.factor(y)
mydata=data.frame(y,x)
waarde=attrEval(as.factor(Ytrain)~.,mydata,estimator="InfGain")
for (j in 1:p) Fwaardes[j]=waarde[j]
return(Fwaardes)
}

#############################################################################
# Now the main program starts.                                              #
#############################################################################

proc1measures=matrix(0,M,7)
proc2measures=array(0,c(K,M,7))
proc3measures=array(0,c(K,M,7))
proc4measures=array(0,c(K,M,7))
proc5measures=array(0,c(K,M,7))
Cpar=1
opt.C=Cpar
avarsel=array(0,c(p,4,M,K))
aavars=array(0,c(4,p,M,K))
aamat=array(0,c(3,p,M,K))
spolaor=array(0,c(K,p,M,K))
```

```
#########################################################################
# Start the loop that repeatedly splits the benchmark dataset into      #
#     training and test parts.                                          #
#########################################################################

for (itel in 1:M) {
      synthdatatrain = synth(N = 80,k = 10,pnoise = 10,q = 6,rho = 0,pvek =
            c(0.4, 0.4, 0.4, 0.4, 0.4, 0.4), Amat = Amat, signal = 10)
      Ytrain = synthdatatrain[[2]]
      Xtrain = synthdatatrain[[1]]
      synthdatatest = synth(N = 10000,k = 10,pnoise = 10,q = 6,rho = 0,pvek =
            c(0.4, 0.4, 0.4, 0.4, 0.4, 0.4), Amat = Amat, signal = 10)
      Ytest = synthdatatest[[2]]
      Xtest = synthdatatest[[1]]
      Ntrain = nrow(Ytrain)
      Ntest = nrow(Ytest)
      p = ncol(Xtrain)
      K = ncol(Ytrain)
      Fmat_svm = matrix(0,Ntest,K)
      Fmat2_svm = matrix(0,Ntrain,K)
      Zfull=matrix(0,nrow=Ntest,ncol=K)
      ranksfull=matrix(0,nrow=Ntest,ncol=K)
      variables=matrix(0,4,p)
      meanfull=matrix(apply(Xtrain,2,mean),ncol=ncol(Xtrain))
      varifull=matrix(apply(Xtrain,2,var),ncol=ncol(Xtrain))
      Xtrainscalesvm=scale(Xtrain)
      Xtestscalesvm=scale(Xtest,center=meanfull,scale=sqrt(varifull))
      findgam=sigest(as.matrix(Xtrain),frac=1,scaled=TRUE)
      opt.gam=findgam[2]
      rbf=rbfdot(sigma=opt.gam)
      gmat=kernelMatrix(rbf,Xtestscalesvm,Xtrainscalesvm)
      gmat2=kernelMatrix(rbf,Xtrainscalesvm,Xtrainscalesvm)

#########################################################################
# Perform binary relevance, with an SVM as base classifier, using all   #
#     the input features.                                               #
#########################################################################
```

```
      for (j in 1:K) {
            Ytrainsvm=Ytrain[,j]
            Ytrainsvm[Ytrain[,j]==0]=-1
            Ytrainsvm=factor(Ytrainsvm)
            svmfit=ksvm(x=Xtrainscalesvm,y=Ytrainsvm,type="C-
                  svc",kernel="rbfdot",kpar=list(sigma=opt.gam),C=Cpar,prob.m
                  odel=FALSE)
            coefsvm=as.matrix(unlist(coef(svmfit)))
            bcoef=unlist(b(svmfit))
            indeks=unlist(SVindex(svmfit))
            coefvol=rep(0,Ntrain)
            coefvol[indeks]=coefsvm
            q3=matrix(t(as.matrix(coefvol))%*%t(gmat),ncol=1)
            fvalues=q3-bcoef[1]
            Fmat_svm[,j]=fvalues
            q3=matrix(t(as.matrix(coefvol))%*%t(gmat2),ncol=1)
            fvalues=q3-bcoef[1]
            Fmat2_svm[,j]=fvalues
      }
fixedt=0.5
print (itel)
labels=threshold(fixedt,Ytrain,Fmat2_svm,Fmat_svm)
Zfull=labels[[1]]
ranksfull=labels[[2]]
a=measures(Ytest,Zfull,ranksfull)
for (j in 1:6) proc1measures[itel,j]=a[[j]]
proc1measures[itel,7]=p

#########################################################################
# Now the different selection procedures start.                         #
#########################################################################

spolaorthreshold=c(0.01,0.03,0.05,0.06,0.07,0.08)
seleksieafvoer=selectionfour(alfa,Bselect,K,Xtrain,Ytrain)
seltelvect=c(1,2,3,4,5,6)
```

```
#########################################################################
# Start the loop that considers different thresholds to declare a       #
#     feature globally relevant.                                        #
#########################################################################

for (seltel in 1:6) {
      labelcut=seltelvect[seltel]
      spolthres=spolaorthreshold[seltel]
      simselvars=matrix(0,4,ncol(Xtrain))
      Fmatsel_svm = matrix(0,Ntest,K)
      Fmat2sel_svm = matrix(0,Ntrain,K)
      Zsel=matrix(0,nrow=Ntest,ncol=K)
      rankssel=matrix(0,nrow=Ntest,ncol=K)
      spolaor[,,itel,seltel]=seleksieafvoer[[2]]

#########################################################################
# Perform probe selection using the correlation coefficient as          #
#     importance measure.                                               #
#########################################################################

            t1=which(apply(seleksieafvoer[[1]][,,1],2,sum)>labelcut)
            if (length(t1)<p) t1=c(t1,rep(0,p-length(t1)))
            if (any(t1>0))   variables[1,]=t1
            if (any(variables[1,]!=0)) {
                  avarsel[variables[1,which(variables[1,]>0)],1,itel,seltel]=
                        1
                  aavars[1,,itel,seltel]=variables[1,]
                  aamat[1,,itel,seltel]=apply(seleksieafvoer[[1]],2,sum)
                  Xtrainsel=Xtrain[,variables[1,which(variables[1,]>0)]]
                  dim(Xtrainsel)=c(Ntrain,length(which(variables[1,]>0)))
                  Xtrainscalesvm=scale(Xtrainsel)
                  Xtestsel=Xtest[,variables[1,which(variables[1,]>0)]]
                  meansel=matrix(apply(Xtrainsel,2,mean),ncol=ncol(Xtrainsel)
                        )
                  varsel=matrix(apply(Xtrainsel,2,var),ncol=ncol(Xtrainsel))
                  Xtestscalesvm=scale(Xtestsel,center=meansel,scale=sqrt(vars
                        el))
                  findgam=sigest(as.matrix(Xtrainsel),frac=1,scaled=TRUE)
```

```
                opt.gam=findgam[2]
                rbf=rbfdot(sigma=opt.gam)
                gmat=kernelMatrix(rbf,Xtestscalesvm,Xtrainscalesvm)
                gmat2=kernelMatrix(rbf,Xtrainscalesvm,Xtrainscalesvm)
                for (j in 1:K) {
                      Ytrainsvm=Ytrain[,j]
                      Ytrainsvm[Ytrain[,j]==0]=-1
                      Ytrainsvm=factor(Ytrainsvm)
                      svmfit=ksvm(x=Xtrainscalesvm,y=Ytrainsvm,type="C-
                            svc",kernel="rbfdot",kpar=list(sigma=opt.gam),C=
                            Cpar,prob.model=FALSE)
                      coefsvm=as.matrix(unlist(coef(svmfit)))
                      bcoef=unlist(b(svmfit))
                      indeks=unlist(SVindex(svmfit))
                      coefvol=rep(0,Ntrain)
                      coefvol[indeks]=coefsvm
                      q3=matrix(t(as.matrix(coefvol))%*%t(gmat),ncol=1)
                      fvalues=q3-bcoef[1]
                      Fmatsel_svm[,j]=fvalues
                      q3=matrix(t(as.matrix(coefvol))%*%t(gmat2),ncol=1)
                      fvalues=q3-bcoef[1]
                      Fmat2sel_svm[,j]=fvalues
                }
                labels=threshold(fixedt,Ytrain,Fmat2sel_svm,Fmatsel_svm)
                Zsel=labels[[1]]
                rankssel=labels[[2]]
                simselvars[1,variables[1,which(variables[1,]>0)]]=1
                proc2measures[seltel,itel,7]=apply(simselvars,1,sum)[1]
                a=measures(Ytest,Zsel,rankssel)
                for (j in 1:6) proc2measures[seltel,itel,j]=a[[j]]
          }

#########################################################################
# Perform probe selection using ReliefF as importance measure           #
#########################################################################

          t2=which(apply(seleksieafvoer[[1]][,,2],2,sum)>labelcut)
          if (length(t2)<p) t2=c(t2,rep(0,p-length(t2)))
```

```
if (any(t2>0)) variables[2,]=t2
if (any(variables[2,]!=0)) {
      avarsel[variables[2,which(variables[2,]>0)],2,itel,seltel]=
            1
      aavars[2,,itel,seltel]=variables[2,]
      aamat[2,,itel,seltel]=apply(seleksieafvoer[[1]],2,sum)
      Xtrainsel=Xtrain[,variables[2,which(variables[2,]>0)]]
      dim(Xtrainsel)=c(Ntrain,length(which(variables[2,]>0)))
      Xtrainscalesvm=scale(Xtrainsel)
      Xtestsel=Xtest[,variables[2,which(variables[2,]>0)]]
      meansel=matrix(apply(Xtrainsel,2,mean),ncol=ncol(Xtrainsel)
            )
      varsel=matrix(apply(Xtrainsel,2,var),ncol=ncol(Xtrainsel))
      Xtestscalesvm=scale(Xtestsel,center=meansel,scale=sqrt(vars
            el))
      findgam=sigest(as.matrix(Xtrainsel),frac=1,scaled=TRUE)
      opt.gam=findgam[2]
      rbf=rbfdot(sigma=opt.gam)
      gmat=kernelMatrix(rbf,Xtestscalesvm,Xtrainscalesvm)
      gmat2=kernelMatrix(rbf,Xtrainscalesvm,Xtrainscalesvm)
      for (j in 1:K) {
            Ytrainsvm=Ytrain[,j]
            Ytrainsvm[Ytrain[,j]==0]=-1
            Ytrainsvm=factor(Ytrainsvm)
            svmfit=ksvm(x=Xtrainscalesvm,y=Ytrainsvm,type="C-
                  svc",kernel="rbfdot",kpar=list(sigma=opt.gam),C
                  =Cpar,prob.model=FALSE)
            coefsvm=as.matrix(unlist(coef(svmfit)))
            bcoef=unlist(b(svmfit))
            indeks=unlist(SVindex(svmfit))
            coefvol=rep(0,Ntrain)
            coefvol[indeks]=coefsvm
            q3=matrix(t(as.matrix(coefvol))%*%t(gmat),ncol=1)
            fvalues=q3-bcoef[1]
            Fmatsel_svm[,j]=fvalues
            q3=matrix(t(as.matrix(coefvol))%*%t(gmat2),ncol=1)
            fvalues=q3-bcoef[1]
            Fmat2sel_svm[,j]=fvalues
```

```
                }
                labels=threshold(fixedt,Ytrain,Fmat2sel_svm,Fmatsel_svm)
                Zsel=labels[[1]]
                rankssel=labels[[2]]
                simselvars[2,variables[2,which(variables[2,]>0)]]=1
                proc3measures[seltel,itel,7]=apply(simselvars,1,sum)[2]
                a=measures(Ytest,Zsel,rankssel)
                for (j in 1:6) proc3measures[seltel,itel,j]=a[[j]]
          }

#########################################################################
# Perform probe selection using information gain as importance measure. #
#########################################################################

          t3=which(apply(seleksieafvoer[[1]][,,3],2,sum)>labelcut)
          if (length(t3)<p) t3=c(t3,rep(0,p-length(t3)))
          if (any(t3>0)) variables[3,]=t3
          if (any(variables[3,]!=0)) {
                avarsel[variables[3,which(variables[3,]>0)],3,itel,seltel]=
                      1
                aavars[3,,itel,seltel]=variables[3,]
                aamat[3,,itel,seltel]=apply(seleksieafvoer[[1]],2,sum)
                Xtrainsel=Xtrain[,variables[3,which(variables[3,]>0)]]
                dim(Xtrainsel)=c(Ntrain,length(which(variables[3,]>0)))
                Xtrainscalesvm=scale(Xtrainsel)
                Xtestsel=Xtest[,variables[3,which(variables[3,]>0)]]
                meansel=matrix(apply(Xtrainsel,2,mean),ncol=ncol(Xtrainsel)
                      )
                varsel=matrix(apply(Xtrainsel,2,var),ncol=ncol(Xtrainsel))
                Xtestscalesvm=scale(Xtestsel,center=meansel,scale=sqrt(vars
                      el))
                findgam=sigest(as.matrix(Xtrainsel),frac=1,scaled=TRUE)
                opt.gam=findgam[2]
                rbf=rbfdot(sigma=opt.gam)
                gmat=kernelMatrix(rbf,Xtestscalesvm,Xtrainscalesvm)
                gmat2=kernelMatrix(rbf,Xtrainscalesvm,Xtrainscalesvm)
          for (j in 1:K) {
                Ytrainsvm=Ytrain[,j]
```

```
                Ytrainsvm[Ytrain[,j]==0]=-1
                Ytrainsvm=factor(Ytrainsvm)
                svmfit=ksvm(x=Xtrainscalesvm,y=Ytrainsvm,type="C-
                      svc",kernel="rbfdot",kpar=list(sigma=opt.gam),C=Cpar,
                      prob.model=FALSE)
                coefsvm=as.matrix(unlist(coef(svmfit)))
                bcoef=unlist(b(svmfit))
                indeks=unlist(SVindex(svmfit))
                coefvol=rep(0,Ntrain)
                coefvol[indeks]=coefsvm
                q3=matrix(t(as.matrix(coefvol))%*%t(gmat),ncol=1)
                fvalues=q3-bcoef[1]
                Fmatsel_svm[,j]=fvalues
                q3=matrix(t(as.matrix(coefvol))%*%t(gmat2),ncol=1)
                fvalues=q3-bcoef[1]
                Fmat2sel_svm[,j]=fvalues
          }
          labels=threshold(fixedt,Ytrain,Fmat2sel_svm,Fmatsel_svm)
          Zsel=labels[[1]]
          rankssel=labels[[2]]
          simselvars[3,variables[3,which(variables[3,]>0)]]=1
          proc4measures[seltel,itel,7]=apply(simselvars,1,sum)[3]
          a=measures(Ytest,Zsel,rankssel)
          for (j in 1:6) proc4measures[seltel,itel,j]=a[[j]]
    }

#########################################################################
# Perform selection using the approach of Spolaor.                      #
#########################################################################

    t4=which(apply(seleksieafvoer[[2]],2,mean)>spolthres)
    if (length(t4)<p) t4=c(t4,rep(0,p-length(t4)))
    if (any(t4>0))   variables[4,]=t4
    if (any(variables[4,]!=0)) {
          avarsel[variables[4,which(variables[4,]>0)],4,itel,seltel]=1
          aavars[4,,itel,seltel]=variables[4,]
          Xtrainsel=Xtrain[,variables[4,which(variables[4,]>0)]]
          dim(Xtrainsel)=c(Ntrain,length(which(variables[4,]>0)))
```

```
        Xtrainscalesvm=scale(Xtrainsel)
        Xtestsel=Xtest[,variables[4,which(variables[4,]>0)]]
        meansel=matrix(apply(Xtrainsel,2,mean),ncol=ncol(Xtrainsel))
        varsel=matrix(apply(Xtrainsel,2,var),ncol=ncol(Xtrainsel))
        Xtestscalesvm=scale(Xtestsel,center=meansel,scale=sqrt(varsel))
        findgam=sigest(as.matrix(Xtrainsel),frac=1,scaled=TRUE)
        opt.gam=findgam[2]
        rbf=rbfdot(sigma=opt.gam)
        gmat=kernelMatrix(rbf,Xtestscalesvm,Xtrainscalesvm)
        gmat2=kernelMatrix(rbf,Xtrainscalesvm,Xtrainscalesvm)
        for (j in 1:K) {
              Ytrainsvm=Ytrain[,j]
              Ytrainsvm[Ytrain[,j]==0]=-1
              Ytrainsvm=factor(Ytrainsvm)
              svmfit=ksvm(x=Xtrainscalesvm,y=Ytrainsvm,type="C-
                    svc",kernel="rbfdot",kpar=list(sigma=opt.gam),C=Cpar,
                    prob.model=FALSE)
              coefsvm=as.matrix(unlist(coef(svmfit)))
              bcoef=unlist(b(svmfit))
              indeks=unlist(SVindex(svmfit))
              coefvol=rep(0,Ntrain)
              coefvol[indeks]=coefsvm
              q3=matrix(t(as.matrix(coefvol))%*%t(gmat),ncol=1)
              fvalues=q3-bcoef[1]
              Fmatsel_svm[,j]=fvalues
              q3=matrix(t(as.matrix(coefvol))%*%t(gmat2),ncol=1)
              fvalues=q3-bcoef[1]
              Fmat2sel_svm[,j]=fvalues
        }
        labels=threshold(fixedt,Ytrain,Fmat2sel_svm,Fmatsel_svm)
        Zsel=labels[[1]]
        rankssel=labels[[2]]
        simselvars[4,variables[4,which(variables[4,]>0)]]=1
        proc5measures[seltel,itel,7]=apply(simselvars,1,sum)[4]
        a=measures(Ytest,Zsel,rankssel)
        for (j in 1:6) proc5measures[seltel,itel,j]=a[[j]]
}
```

```
#############################################################################
# End the loop that considers different thresholds to declare a             #
#     feature globally relevant.                                            #
#############################################################################
            }
#############################################################################
# End the loop that repeatedly splits the benchmark dataset into            #
#     training and test parts.                                              #
#############################################################################
}

library(reshape2)
write.table(proc1measures, "C:\\...\\results.full.probe.txt")
write.table(melt(proc2measures), "C:\\...\\results.cor.probe.txt")
write.table(melt(proc3measures), "C:\\...\\results.reliefF.probe.txt")
write.table(melt(proc4measures), "C:\\...\\results.IG.probe.txt")
write.table(melt(proc5measures), "C:\\...\\results.spolaor.probe.txt")
```