# Extracting Failure Modes from Unstructured, Natural Language Text

by

Francina Malan

*Thesis presented in fulfilment of the requirements for the degree of Master in Industrial Engineering in the Faculty of Engineering at Stellenbosch University*

Supervisor: Dr. J.L. Jooste

March 2020

# Declaration

By submitting this thesis 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.

Date: .........March 2020..............

# Abstract

## Extracting Failure Modes from Unstructured, Natural Language Text

F. Malan

*Department of Industrial Engineering,*
*University of Stellenbosch,*
*Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (Industrial)

March 2020

This thesis investigates whether text mining (and the related fields of machine learning and natural language processing) can be used to extract useful information, specifically failure modes, from the low quality, unstructured text records available in industry.

Failure data, and particularly information about failure modes, is imperative for good asset management, but frequently goes underutilised because it is buried in unstructured text which is not amenable to traditional analytics, but is too resource intensive to process manually. While the ideal solution would be to improve the information management system to prevent the collection of such data, this only addresses the quality of future data while years of historic data will then be lost.

Several authors have acknowledged the prevalence of text-based maintenance records, identifying both the potential value and problems in utilising this data, with many suggesting some form of text mining as a possible solution. Within this and related fields, there is a gap between the academic and industry focussed literature. This pertains to both the scarcity of industry (and especially maintenance specific) research and the inadequate attention given to the theoretical basis of these fields in the available industry literature. The biggest concern pertains to the violation of the independent, identically distributed (IID) assumption in maintenance data and the impact this has on the validity of various evaluation schemes. Other concerns regard the optimisation of preprocessing parameters and the evaluation metric used to assess performance.

This project was completed within the CRISP-DM framework. For the research objectives, both the more practical industry-focussed studies and the more theoretical, academic studies were investigated. In the experimental component, two families of algorithms were evaluated, namely Support Vector Machines and Naïve Bayes. The focus was on the validity of the modelling and evaluation process based on problems identified in literature. Noteworthy aspects of this procedure include using a blocked cross-validation as the outer, evaluation loop of a nested cross-validation to account for the IID violation and to prevent the over-optimisation that can occur from single-loop cross-validation.

The most important contribution of this work is the experimental design which consolidates multiple validity concerns raised in academic literature but receive limited attention in industry. In particular, it addresses the violation of the IID assumption in standard cross-validations (Bergmeir and Benitez, 2012), the importance of including preprocessing into the model optimisation (Krstajic *et al.*, 2014), the high potential of randomised search optimisation (Bergstra and Bengio, 2012) and the different formulations of the cross-validated F-score (Forman and Scholz, 2010). The recommendations made by authors investigating these issues in isolation were combined to form the experimental design. It is however worth noting that the methodological conclusions made in this study are based on the evaluation of a single dataset and is not necessarily indicative of the general behaviour.

The project concludes that while text mining offers a viable solution for the identified problem, doing so is not a trivial process and would require substantial commitment from organisations wishing to utilise their data.

# Opsomming

## Die Bepaling van Falingsmodusse Vanuit Ongestruktureerde, Natuurlike Taal Teks

*("Extracting Failure Modes from Unstructured, Natural Language Text ")*

F. Malan

*Departiment van Bedryfs Ingenieurswese,*
*Universiteit van Stellenbosch,*
*Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MIng (Bedryfs)

Maart 2020

Hierdie tesis ondersoek die moontlikheid of teks ontginning (en die aanverwante velde van masjienleer en natuurlike taal prosessering) gebruik kan word om bruikbare inligting (spesifiek falings modusse) te bekom van die lae gehalte, ongestruktureerde teks rekords wat in die industrie beskikbaar is.

Falings data, en spesifiek inligting rakende falings modusse, is onontbeerlik vir goeie batebestuur, maar word dikwels onbenut omdat dit in teks formaat versamel word wat nie geskik is vir tradisionele data-analises nie, maar ook te hulpbronintensief is om met die hand te verwerk. Alhoewel die ideale oplossing sou wees om die inligting-bestuur-sisteem te verbeter om te voorkom dat sulke data versamel word, sal dit slegs die kwaliteit van toekomstige data aanspreek terwyl jare van historiese data verlore sal gaan.

Verskeie skrywers bevestig dat teksgebaseerde instandhoudings-rekords algemeen is, en identifiseer beide die potensiële waarde en die probleme in die gebruik hiervan, wat baie lei tot die voorstel om teks ontginning te gebruik. Binne hierdie velde is daar 'n gaping tussen die akademiese en industrie-gefokusde literatuur. Dit het betrekking tot die skaarsheid van industrie (en veral instandhoudings-spesifieke) navorsing en die onvoldoende aandag wat aan die teoretiese basis van hierdie velde gegee word in die beskikbare industrie-literatuur. Die grootste bekommernis is die oortreding van die IID aanname in instandhoudings data en die impak wat dit op evaluasie skemas het. Ander bekommernisse is die optimalisering van vooraf-prosessering parameters, en die evaluasie-maatstaf wat gebruik word.

Hierdie projek is gedoen binne die CRISP-DM raamwerk. Beide die meer praktiese industrie-gefokusde studies en die meer teoretiese, akademiese studies is ondersoek. In die eksperimentele komponent is twee algoritme klasse ge-evalueer: Support Vector Machines en Naïve Bayes. Die fokus was op die geldigheid van die modellering- en evaluasie proses, gebaseer op probleme soos geïdentifiseer in literatuur. Opvallende aspekte in hierdie prosedure is die gebruik van geblokkeerde kruis-verifiëring in die buitenste evaluasie lus van 'n geneste kruis-verifiëring om rekenskap te gee van die IID skending en te voorkom dat oor-optimalisering kan gebeur van enkel lus kruis-verifiëring.

Die mees belangrike bydrae van hierdie werk is die eksperimentele ontwerp wat veelvoudige geldigheids-bekommernisse konsolideer, wat reeds in akademiese literatuur genoem is, maar weinig aandag kry in die industrie. In besonder adresseer dit die oortreding van die IID aanname in standaard kruis-verifiërings (Bergmeir en Benitez, 2012), die belangrikheid daarvan om vooraf-prosessering in te sluit in die model optimalisering (Krstajic *et al.*, 2014), die hoë potensiaal van lukrake soek-optimalisering (Bergstra en Bengio, 2012) en die verskillende formulerings van die kruis-geverifieerde F-telling (Forman en Scholz, 2010). Die aanbevelings van skrywers wat hierdie probleme in isolasie nagevors het, word gekombineer om die eksperimentele ontwerp te vorm. Dit is egter nodig om te noem dat die metodologiese bevindinge uit hierdie studie gebaseer is op die evaluasie van 'n enkele datastel en nie noodwendig aanduidend is van algemene gedrag nie.

Die projek se bevinding is dat alhoewel teks ontginning 'n oplossing bied vir die geïdentifiseerde probleem, dit nie ’n maklike proses is nie en vereis substansiële toewyding van organisasies wat hul data wil benut.

# Acknowledgements

I would like to express my sincere gratitude towards Pragma who funded this research and in particular Messrs M. Steenkamp and B. Uys who provided great assistance. Also, I would like to thank my study leader Dr J. L. Jooste for his guidance and patience.

This project would not have been possible without the enduring support and encouragement from my family and friends. I am thankful to all of you and especially my parents, sister, extended Stellenbosch family and Jaco.

# Contents

# List of Figures

# List of Tables

# Nomenclature

## Variables

| | |
|---|---|
| F | F1-score (Harmonic mean between precision and recall) |
| FP | False Positives |
| FN | False Negatives |
| TN | True Negatives |
| TP | True Positives |
| V | Corpus vocabulary |
| $X$ | Input Data (Documents) |
| $y$ | Output Data (Labels) |

## Subscripts

| | |
|---|---|
| AGG | Aggregated Mean |
| AVG | Averaged Mean |
| geom | Geometric Mean |
| $i$ | Matrix row coordinate (document number) |
| $j$ | Matrix column coordinate (word number) |

| | |
|---|---|
| macro | Macro-Averaged |
| micro | Micro-Averaged |

# List of Acronyms and Abbreviations

### Abbreviations

| | |
|---|---|
| AM | Asset Management |
| BD | Big Data |
| BOW | Bag-Of-Words |
| CM | Condition Monitoring |
| CV | Cross Validation |
| DF | Document Frequency |
| DM | Data Mining |
| DTM | Document Term Matrix |
| EOI | Entities of Interest |
| OEM | Original Equipment Manufacturers |
| FM | Failure Mode |
| IDF | Inverse Document Frequency |
| IID | Independent, Identically Distributed |
| KDDM | Knowledge Discovery and Data Mining |

| | |
|---|---|
| ML | Machine Learning |
| NCV | Nested Cross Validation |
| NLP | Natural Language Processing |
| OVO | One-vs-One |
| OVR | One-vs-Rest |
| POS | Parts of Speech |
| SME | Subject Matter Experts |
| SVC | Support Vector Classifier |
| SVM | Support Vector Machine |
| TF | Term Frequency |
| TFIDF | Term Frequency Inverse Document Frequency |
| TM | Text Mining |
| VSM | Vector Space Model |
| WO | Work Order |

**Acronyms**

| | |
|---|---|
| CMMS | Computerised Maintenance Management System |
| CRISP-DM | CRoss-Industry Standard Process for Data Mining |
| FMEA | Failure Modes and Effects Analysis |
| GFMAM | Global Forum for Maintenance and Asset Management |
| IAM | Institute of Asset Management |

| | |
|---|---|
| ISO | International Standards Organization |
| NLTK | Natural Language Toolkit |
| PAS | Publicly Available Standard |
| Scikit-learn | SciPy Toolkit - Scientific and technical computing Python library |

## Models

| | |
|---|---|
| bernNB | Bernoulli Naïve Bayes |
| multiNB | Multinomial Naïve Bayes |
| linSVC | LIBLINEAR SVC implementation on scikit-learn |
| SVC | LIBSVM SVC implementation on scikit-learn |

# Chapter 1

# Introduction

This chapter introduces the research undertaken starting with a brief overview of the background and motivation of the study. After this, the problem statement is presented in Section 1.2 describing the gaps in literature that require further investigation. Next, Section 1.3 positions the study within these gaps by defining a research question and project objectives to contribute towards the current body of knowledge that address the problem statement. This is followed by a discussion of the project limitations and delimitations in Section 1.4 after which the chapter concludes with an outline of the remainder of the document.

## 1.1 Introduction

Assets are the foundation of value creation in any business be it tangible assets such as plant equipment or intangible assets such as employee skills. In the increasingly competitive business environment of today, it is becoming more important than ever to maximise the value that can be extracted from these assets for the minimum input. For this reason, a holistic approach to asset management (rather than the traditional maintenance based focus) is becoming more and more popular. An important part of asset (and all other types of) management is having up-to-date, accurate and comprehensive information of a good quality (GFMAM, 2014). As Peter Drucker, a famous management consultant, once said, *"You can't manage what you don't measure."* (McAfee and Brynjolfsson, 2012).

One of the most important pieces of information to "measure" in asset management is the failure data (and particularly the failure modes) of assets. Without proper knowledge of the actual failure modes, activities such as Failure Mode and Effects Analysis (FMEA), Root Cause Analysis (RCA) or strategies such as Six Sigma, Total Quality Management (TQM) or Reliability Centred Maintenance (RCM) are not evidence-based and at risk of having limited success. In fact, according to Reeve (2016) this data is essential for even the most fundamental aspects of strategic asset management. This is confirmed by the various asset management (AM) resources such as ISO 55000, the AM Landscape and the AM Anatomy which emphasise the importance of evidence based, data-driven decision making and that the availability of failure data (and its analysis) can greatly impact the ability of organisations to optimise the cost, risk and performance trade-off which is the ultimate business objective (ISO 2014; GFMAM 2014; IAM 2014).

However, while Computerised Maintenance Management Systems (CMMS) have made it easy to collect data, ensuring the quality of this data remains a challenging issue (Reeve, 2016). Poor data quality can lead to great economic loss since data-driven decisions are only as good as the data used to make them (Woodall *et al.*, 2015). Data quality does not only pertain to correctness (or accuracy) of the data. Bad information management practices can also result in data which may be accurate, but is not amenable to computerised analyses (Reeve, 2016). As Woodall *et al.* (2015) points out, if the data is not accessible, or requires an impractical amount of effort to utilise, all other quality considerations are irrelevant and the data is essentially worthless (Woodall *et al.*, 2015). This is especially prevalent in low maturity organisations which are typically characterised by departmentally fragmented silos and poor line of sight (Woodall *et al.* 2015; IAM 2014). Because the data is not collected by the end-user, it is often initiated without clear purpose of use (collecting for the sake of collecting) and in the wrong format (easy to record, but difficult to analyse) (Reeve 2016; Woodall *et al.* 2015; Devaney *et al.* 2005). Operation and maintenance staff will typically store failure data in free-format, natural language text which does not allow for traditional data-science based failure analyses (Reeve 2016; Uz-Zaman *et al.* 2015; McKenzie *et al.* 2010). These are among the reasons why up to 70% of organisations do not perform even basic failure analysis on their data and instead rely on expert intuition and best-guesses rather than fact-based decision making (Reeve 2016; Edwards *et al.* 2008).

The ideal solution would be to improve the entire information management system (Reeve, 2016). Standards such as ISO 8000 (Data Quality), ISO 50000 (Asset Management) and frameworks such as the AM Landscape are only a few examples of the many resources available that address the quality of data and maintenance data specifically. A common element advocated by these resources is the need to introduce a coding or categorisation system that records failure data (and other important information) in a structured manner that is well-suited to automated analyses (ISO 2014; ISO 2011; GFMAM 2014).

However, major system overhauls require significant initial investments of time, money and expertise as well as sustained motivation and discipline from all stakeholders and a continued commitment by management to provide sufficient resources (ISO 2014; ISO 2011; Mobley 2002). Low maturity companies especially struggle with this as they are often caught up in fire-fighting immediate problems and prioritise short-term operational needs over long-term, focussed improvement processes that do not yield immediate returns (Mahmood *et al.*, 2015). If not executed properly an organisation can easily slip back to their previous, or a different but equally undesirable, state (BSI, 2008). Furthermore, improved information management strategies can only help improve the quality of data collected in the future whereas many companies have years of unused, potentially valuable, historical data that will then go to waste (Reeve 2016; Devaney *et al.* 2005).

This is not a problem isolated to asset management. It has been estimated that up to 80% of all organizational data is in text format which requires time-consuming and labour-intensive manual processing which frequently leads to the data being either ignored or ineffectively utilised (Singh and Raghuvanshi 2012; Kobayashi *et al.* 2018). Furthermore, the data creating capabilities of many industries have far surpassed the ability to handle it with many companies hoarding vast amounts of data they will never use (Russom, 2011). Not only do they not benefit from this data, but it has been recognised that excessive amounts of information may even have a negative organisational impact because it distracts from the main issues, delays decision making and can lead to wrong conclusions if incorrectly utilised (Wuest *et al.*, 2016). In fact, this overwhelming flood of information, or "Big Data" as it is called, has led to 30% of companies viewing data as a problem rather than the opportunity it could be according to the Transforming Data With Intelligence (TDWI) Best Practices Report (Russom, 2011).

There are no threshold values for when data becomes Big Data. Rather, it is any data that cannot be processed using traditional data analytics and is typically characterised by the four V's: *Volume*, *Velocity*, *Veracity* and *Variety* (Sathi, 2012). In the analysis of asset (maintenance) related data, variety is an especially big issue. Variety concerns the difficulties involved in treating the broad range of data formats that contain valuable information but do not come in structured, easy-to-analyse datasets; such as the text-based maintenance records collected by many organisations.

The analysis of unstructured data, and specifically natural language text, is one of the fundamental focusses of Big Data analytics (Chen *et al.*, 2014). As such, Big Data analysis techniques such as text mining could potentially be used to interpret asset related data. Text mining concerns the automated, or semi-automated (computer-assisted) extraction of useful knowledge from text data (Wachsmuth, 2015). While there is some contention in literature as to the exact definition and scope of text mining, it is generally accepted that it is a multidisciplinary field that draws heavily from the fields of machine learning (ML) and natural language processing (NLP) with some seeing it as a branch or sub-field of data mining and others as a distinct but correlated, sibling field (Dhanrajani and Gosh 2008; Witten *et al.* 2011).

For this thesis, text mining is used to mean text data mining with data mining defined as the exploration and analysis of large quantities of data by automatic or semi-automatic means with the purpose of obtaining useful knowledge (paraphrased from the original definition by Berry and Linoff (1997) cited in Bastos *et al.* (2014)). Machine learning can be seen as the technical basis of data (or text) mining and refers to the area of artificial intelligence that allows computers to "learn" from data rather than being explicitly programmed (Bastos *et al.* 2014; Witten *et al.* 2011). Natural language processing, which is a sub-field of artificial intelligence and computational linguistics, is the broad term for the integration of human language and computers that is concerned with converting natural language to machine-readable, structured data (Dhanrajani and Gosh, 2008). This means that NLP can be both a text mining tool and a text mining outcome.

Despite the established need of text mining in industry, its uptake has been slow for various reasons (Kobayashi *et al.*, 2018). Firstly, while there exists a considerable amount of text mining and machine learning literature, it suffers from a lack of standardisation in terms of both methodology and terminology (Moreno-Torres *et al.*, 2012). This is partly due to being relatively new disciplines, and partly due to simultaneously developing from multiple different fields with different objectives, terminologies and standard practices. The wide range of algorithms, theories and diverging opinions which are documented in literature poses an especially big problem for novice practitioners (which industry, and maintenance, practitioners are likely to be) and represents a barrier for industry adoption (Wuest *et al.*, 2016). This is further exacerbated by the inconsistent use of terminology with different terms used to describe the same concept, or even worse, using the same term to mean different things. Not only does this make it difficult to find relevant literature, but it also hampers effective knowledge sharing, fair comparisons between studies and the identification of best practices (Moreno-Torres *et al.*, 2012).

Furthermore, according to Kobayashi *et al.* (2018) slow organisational uptake is also due to the fact that only a small portion of the available literature is industry focussed. The majority is targeted towards academic researchers with a skewed focus on technical details and little regard for more practical concerns. In particular, they emphasise the importance of not only assessing, but also being able to demonstrate the validity of text mining outcomes for industry applications. Since business decisions based on these outcomes will be highly dependent on their reliability, organisational uptake is likely to hinge on the acceptability of these results to industry stakeholders (Kobayashi *et al.*, 2018).

## 1.2 Problem Statement

Failure data, and particularly information about failure modes, is imperative to good asset management but frequently goes unutilised due to poor information management practices. An increased awareness of the importance of information-driven asset management has led to many businesses hoarding years of maintenance records with this objective in mind. However, because they typically collect failure data in the form of unstructured, natural language text with little to no quality measures in place, this data cannot be processed using traditional data analytics but require time-consuming and labour-intensive manual processing which few can afford (McKenzie *et al.* 2010; Chen and Nayak 2007; Rajpathak and De 2016; Devaney *et al.* 2005). This results in having to rely on intuition-based, "best-guess" decision making rather than a facts-based approach (Edwards *et al.*, 2008). The utilisation of such unstructured data sources is a typical application area of Big Data. For this reason, it has been suggested in literature that Big Data techniques such as text mining and machine learning could be viable methods for the analysis of maintenance records (Section 4.1).

Both the potential value and the problems in using the unstructured, free-text portions of maintenance records is well-documented in literature. Several authors confirm not only the prevalence of text-based maintenance records, but also the problematic consequences this holds for industry including the loss of value-adding failure data (McKenzie *et al.* 2010; Chen and Nayak 2007; Rajpathak and De 2016; Devaney *et al.* 2005). The more generic problem of utilising text data effectively is well-documented beyond the maintenance domain as well and has been the focus of much research effort, particularly in the text mining and machine learning literature which have made great advances in recent years. However, Kobayashi *et al.* (2018) points out that this literature is dominated by academia and identifies the need for more industry-focussed research. In particular, they identify validity assessment as a critical research area for industry since business decisions depend on the reliability of text mining outcomes making it an important enabler for organisational uptake (Kobayashi *et al.*, 2018).

A contributing factor to the limited amount of industry focussed literature, is the lack of industry provided training data due to what Kostoff (2005) calls the "*incentives to conceal rather than reveal*". Companies are seldomly willing to put their data in the public domain, wary of sharing data or research with potential competitors. This is especially true in the maintenance domain where data contains details of product and operational problems that organisations little want to advertise (Kostoff, 2005). This means that even fewer studies have been done specific to the maintenance domain which present several unique challenges not typically addressed in text-mining literature such as short document length and highly technical, non-standard vocabularies (McKenzie *et al.* 2010; Edwards *et al.* 2008) (Section 4.1.1). Of those that were found and are discussed in Section 4.1, some like Marzec *et al.* (2014) have produced promising results. However, many have found that several NLP and machine learning methods, which are usually effective for text classification, do not perform well on maintenance records and require further study (McKenzie *et al.* 2010; Edwards *et al.* 2008; Chen and Nayak 2007).

The shortage of labelled training-data has also contributed towards the fairly limited success in the maintenace domain as many researchers have been forced to either label data themselves (as opposed to subject matter experts) such as Edwards *et al.* (2008), or to perform unsupervised learning on the unlabelled data such as Chen and Nayak (2007). Others like Devaney *et al.* (2005) do not even have unlabelled data available and provide only hypothetical analyses. As such, there exists a need to investigate the interaction between dataset size and classifier performance to guide future research and data collection efforts and to address the gap in literature concerning supervised classifiers trained on larger maintenance datasets labelled by subject matter experts (SME).

In fact, there is a generally inadequate consideration of the theoretical issues underpinning the application of machine learning in the maintenance-specific literature. Looking beyond the maintenance domain at the broader machine learning literature (text and otherwise) it seems that there is a considerable gap between the industry-focussed, practical application studies and the more academia-focussed, theoretical studies (Kobayashi *et al.*, 2018). While the former often pays little attention to the theoretical underpinnings of the techniques and methodologies used (which calls in to question the validity of their results), the latter typically uses either synthetic or benchmark data and pays little heed to the practical constraints of reality (which calls into question the applicability of their results). Even validity concerns, which Kobayashi *et al.* (2018) identifies as especially important for business implementations, are often only fleetingly addressed and is not up to par with the studies in the academic domain. Both the industry and academic approaches have merit, but there is a need for research that lies somewhere in the middle of these two extremes. The gap is particularly severe for the maintenance domain, perhaps due to the extremely limited amount of studies compared to even the already limited broader industry focussed literature. Accordingly, there exists a need to address the ML theoretical considerations in the maintenance domain to align it with the progress made in the academic, and to a lesser extent, the domain independent, industry literature.

Furthermore, there is a lack of sufficient documentation in the broader machine learning literature with many authors considering precise implementation details as too tedious for publication (Blamey *et al.*, 2012). Due to the wide array of tools available to machine learning, there is no obvious way of determining which were used making it difficult to study the interaction between different combinations of methods and, as Blamey *et al.* (2012) points out, makes replication almost impossible. The available literature also suffers from a lack of standardisation in terms of both methodology and terminology which presents not only a barrier for industry adoption, but also for effective knowledge sharing between experts in the field (Moreno-Torres *et al.* 2012; Wuest *et al.* 2016). While several authors have addressed these inconsistencies in an attempt to create a unifying framework, these are often only focussed on specific applications such as the number of folds in cross-validation (Anguita *et al.*, 2012) or the different ways to average the F-score (Forman and Scholz, 2010) with the results scattered across literature. There is room for a more comprehensive framework which sacrifices detail for broadness to address all implementation needs. Kobayashi *et al.* (2018) addresses this gap by providing a comprehensive tutorial document specifically tailored towards business applications that covers the whole scope of text-mining classification. While this provides an invaluable starting point, it neglects some important theoretical issues that have been identified in the non-industry literature such as data drift.

## 1.3 Research Question and Project Objectives

The main research question of this study, which is also reflected in the document title, is:

*Can text mining (and the related fields of machine learning and natural language processing) be used to extract useful information, specifically failure modes, from the low quality, unstructured text maintenance records which are typically available in industry?*

A wide range of project objectives were identified to address the knowledge gaps identified in the problem statement, and most importantly, to answer the research question above. This includes both research (1-5) and experimental (6-9) objectives which are summarised below. The chapters addressing these objectives are indicated in parentheses.

The project objectives are:

1. To provide an overview of the context and significance of the issues in asset management data (Chapter 2).

2. To investigate the suitability of text mining (and machine learning) to this problem and compare it with alternatives proposed in literature (Chapters 2, 4).

3. To provide an overview of the most important concepts and terminology of text mining and machine learning to serve as theoretical framework for both the experimental and research component of the study (Chapter 3).

4. To investigate both the domain specific and domain independent text mining literature to:

    - Identify the particular challenges of the maintenance domain (Chapters 2, 4),

- Establish the specific range of methods and techniques applicable to the research question (Chapters 3, 4),
- Investigate the strengths and weaknesses of the current domain literature to identify the most important research areas (Chapter 4),
- Evaluate the state of the domain specific research with respect to that of the domain independent literature (Chapter 4).

5. To identify the most important research areas from the broader machine learning literature to be addressed in the experimental analysis and future research (Chapter 4).
6. To perform an experimental analysis on real world data to evaluate the research question with respect to failure mode extraction (Chapters 5, 6, 7).
7. To evaluate the experimental analysis used to achieve the project objectives and identify potential errors and areas of improvement (Chapters 7, 8).
8. To address the methodological concerns identified from literature with specific focus on validity through the use of, among other things, comparative experiments (Chapter 8).
9. To compare the experimental results with that found in literature and draw conclusions to guide future research (Chapters 8, 9).

## 1.4 Limitations and Delimitations

It is important that research is viewed within the context of the constraints in which it was completed. This includes both the self-imposed restrictions selected to focus the research (delimitations) and those beyond the researcher control (limitations).

It was decided to restrict the study to the open-source, off-the-shelf resources available in the Python environment. Python was chosen for its strong technical capabilities, wide choice of libraries, excellent documentation, active online community and human readable code which makes it one of the easiest languages to learn (Pedregosa *et al.*, 2011). It is also the most popular programming language for analytics, data science and machine learning according to a 2016-2018 survey performed by KDnuggets (Piatetsky, 2018). While this does pose some restrictions on the methods that could be used in this study, this delimitation was deemed negligible due to the large amount of resources available in this environment. In particular, the Python machine learning library, Scikit-learn, was used extensively as it provides state-of-the art implementations of a broad range of ML algorithms while maintaining an easy-to-use interface and excellent documentation making it highly relevant for use by non-specialists outside of computer science (in both industry and research) (Pedregosa *et al.*, 2011).

The use of open-source software also make the results more accessible and relevant to larger audiences than would be the case for expensive commercial solutions. This is in line with current trends in both industry and research where four out of the top five most widely used analytics, data mining and Big Data resources are open-source according to a survey reported on by Chen *et al.* (2014). Other than the obvious cost benefit of open-source software, there are also a few other factors that make it an attractive option.

Commercial software has the advantage of being developed by professionals and sometimes offer more specialised tools and support as well as a friendly graphical interface (Cortez, 2010). However, the involvement of external, third-party tools and services increase potential data safety risks and the licensing structure used by many commercial packages creates a continued reliance on a third-party who may go out of business or might not scale well with changing business needs (Chen *et al.* 2014; Maimon and Rokach 2010). Open-source software typically has a steeper learning curve, but once mastered the user has much more control and understanding of the results. It is almost entirely self-reliant and due to its flexible and extensible nature it can be highly customised to suit and evolve with changing individual needs. Because it can leverage the contributions of a much wider range of practitioners, new methods and advances are often more quickly integrated than in commercial tools allowing open-source projects to be at the forefront of development (Cortez, 2010).

Only off-the-shelf applications were considered as this drastically reduces the technical skill level required making it more feasible for industry practitioners who might not have the programming or mathematical knowledge to develop these from first principles. The standardised format of these modules also allow for easier collaboration and maintenance of the resulting data systems. Using off-the-shelf tools is commonly advocated in literature as there is no benefit in every practitioner reinventing the wheel. In fact, Alpaydin (2010) recommends using as much as possible code from reputable machine learning libraries as they are better tested, more reliable and have been optimised for computing performance. This also facilitates replication and ensures that results are not a consequence of software bugs (Alpaydin, 2010).

The final delimitation was the decision to restrict the study to supervised classification using the traditional bag-of-words model as the problem statement in Section 1.2 is well-suited to this approach. While other approaches may have potential too, it was necessary to narrow the scope of this study to a manageable level.

The biggest limitation to this study concerns the nature of the dataset used in the experimental analysis. Since it was desirable to use industry data, both the quality and quantity are largely beyond the researcher's control. An industry partner made a large sample of historical maintenance records available to this study which has several quality issues pertaining to both the original creation and later annotation (labelling) of these records. While these quality issues are more significant than what is typically addressed in text mining research, it is consistent with that found in the domain specific literature.

The quality issues pertaining to data creation is desirable as this reflects the reality of industry applications and is preferable to using artificially high quality or non-relevant data such synthetic or benchmark datasets. The annotation quality poses a bigger concern as it provides the upper limit of performance that can be achieved by automated means and if done improperly, can artificially limit the learnability of the data (Mozetic *et al.*, 2016) (Section 3.2).

Only a small portion of the available dataset has failure mode labels which were assigned by subject matter experts for various other purposes throughout the data collection period. From the exploratory analysis in Section 6.2 it is clear that these are not of a desirable quality. Unlike record creation, the annotation process is not restricted by the operating environment and should form part of the research effort so that it can be carefully controlled to enable maximum performance. However, contrary to generic applications such as sentiment analysis, the labelling must be done by subject matter experts which were not available due the academic origin of this study. The preferable annotation process that should be used for industry applications to prevent this limitation is discussed briefly in Section 3.2, but for this study it could only be mitigated by careful data selection to discard the worst quality records. The study was further limited by the lack of SME involvement in the overall experimental analysis which is likely to benefit significantly from domain expertise (Witten *et al.*, 2011).

These limitations position the study somewhere between the studies such as Marzec *et al.* (2014) who had extensive SME availability for both the labelling and analysis, and the studies such as Edwards *et al.* (2008) who labelled the data themselves with no SME involvement; and ahead of studies such as Devaney *et al.* (2005) who had no data, labelled or unlabelled. While the data quantity also poses a limitation on the study, it is well beyond that considered in much of the domain specific literature such as Chen and Nayak (2007), Edwards *et al.* (2008), McKenzie *et al.* (2010) or Uz-Zaman *et al.* (2015).

Due to the propriety nature of the data, only the results could be published which poses a limitation on the repeatability of this study. This limitation is faced by all the domain specific studies found and is an unfortunate reality of most industry focussed research. Repeatability was ensured as far as possible by using open-source and off-the-shelf tools and through meticulous documentation of the experimental details as per the reccomendations by Alpaydin (2010). However, until domain specific benchmarks are made available this limitation is likely to persist.

In addition to the data quality, further limitations arise as a result of the assumptions made about the nature of the data. There is evidence of chronological distribution changes (data drift) in the dataset which is a clear violation of the independent, identically distributed (IID) assumption made by the majority of machine learning methods (Section 3.1). Further simplifying assumptions made by the various estimators such as the Naïve Bayes conditional independence assumption or even the common bag-of-words (BOW) assumption are also violated in practice.

Violation of these assumptions mostly affect the modelling process leading to less effective classifiers than would be the case if they held. While this is not ideal, it is an acceptable (and unavoidable) limitation of machine learning. A much bigger concern is that most of the evaluation schemes also make the IID assumption meaning that its violation can impact the validity of the results. This was minimised by implementing what Bergmeir and Benitez (2012) calls blocked cross-validation rather than the traditional random or stratified cross-validation to emulate an out-of sample testing procedure. However, this is not a perfect method and some error will invariably persist. Due to the random uncertainty prevalent in both the training and testing of the algorithm, the true model performances can only ever be approximated and all results given in Chapter 7 must be seen as an imperfect estimate. This is not unique to this study but is an inherent limitation of machine learning which would be true even if the IID assumption held. As the statistician George Box famously said, *"All models are wrong, but some are useful"* (Box, 1979).

## 1.5 Thesis Outline

Chapter 1 introduces both the topic under investigation and the study performed. It touches on the background and motivation of the study and formulates the topic into a research question and nine project objectives. It discusses both the inherent limitations of the study and the deliberate restrictions used to focus the research and scope. It concludes with this document outline.

The literature review spans three chapters. The first, Chapter 2, provides the context, background and motivation for the study and can be considered part of phase one in the CRISP-DM methodology (introduced in Chapter 5). It addresses the first two project objectives identified in Section 1.3.

The second literature review chapter, Chapter 3, provides the theoretical background of the other CRISP-DM steps focussing on the third research objective. It provides an overview of the fundamental concepts, terminology and algorithms of machine learning applied to the text domain.

The final literature review chapter, Chapter 4, reviews the related literature both in and out of the maintenance domain. It addresses the fourth research objective to gain insight into the current state of research as it pertains to solving the research question and how it fits into the broader machine learning literature.

Chapter 5 introduces the research methodology followed for the experimental analysis, namely the Cross-Industry Standard Process for Data Mining (CRISP-DM). It starts by motivating the selection of CRISP-DM followed by a brief discussion of the six phases contained in this methodology and the treatment of each in this document.

Objectives six to eight are addressed in the empirical analysis which spans three chapters. The first of these, Chapter 6, starts by establishing the context through the Business and Data Understanding (first two phases of CRISP-DM) followed by the experimental design created according the the Data Preparation, Modelling and Evaluation principals of CRISP-DM with specific focus on validity (as per the eighth objective).

The results of this analysis, representing the outcome of the sixth objective, are discussed in Chapter 7 and compared to a random baseline to get a better indication of the significance of the performance.

The final chapter of the empirical analysis, Chapter 8, presents a methodological evaluation to confirm the validity of the experimental process used to obtain the results as per the seventh and eight objectives. The outcomes of this validity assessment are also used to better relate the results with those in literature that used different experimental procedures as per the ninth objective.

The thesis concludes with Chapter 9 which summarises the outcomes of this study; specifically with regard to the accomplishment of the project objectives and answering of the research question identified in Section 1.3. It acknowledges the areas of success and failure of this study and concludes with recommendations for future research.

The appendix provides more detailed results of the experimental analysis. The list of references is included at the end of the document.

# Chapter 2

# Literature Review: Background and Motivation

The following three chapters present a literature review that explores the background and motivation of the study (Chapter 2), introduces the theoretical framework (Chapter 3), and evaluates similar research (Chapter 4) in completion of the first five project objectives identified in Section 1.3.

In this chapter the first two project objectives are addressed; namely to explore the context and significance of the issues in asset management data, and to investigate the suitability of text mining to address this compared with alternatives proposed in literature. It starts with an overview of asset management and the importance of fact-based, information-driven decision making. The problem area is explored in more detail to identify possible causes and to determine the severity of the problem in terms of both prevalence and impact. Next, this maintenance-specific problem is aligned with the broader issue of Big Data, followed by an introduction to the related fields of text mining, machine learning and natural language processing which have shown great promise in Big Data analyses. It ends with a discussion of the alternative approach recommended in literature; its advantages, disadvantages and the reasons for going the text mining route.

This chapter can be considered part of the first phase of the CRISP-DM methodology discussed in Chapter 5, namely Business Understanding. However, rather than focussing on one particular organisational context, it develops a more generic, *domain* understanding of the broader asset management industry which was considered to be a more appropriate, generalisable outcome for the academic nature of this project.

## 2.1 Asset Management

ISO 55000 defines assets as an *"item, thing or entity that has potential or actual value to an organization"* (ISO, 2014). In the increasingly competitive business environment of today, it is becoming more important than ever to maximise the value that can be extracted from these assets for the minimum input. For this reason, a holistic approach to asset management (rather than the traditional maintenance based focus) is becoming more and more popular (IAM, 2014). Where pure maintenance is restricted to keeping physical equipment operational, asset management (AM) comprises all *"the coordinated activity of an organization to realize value from its assets"* and involves balancing the conflicting objectives between cost, risk and performance (ISO, 2014).

According to Kans and Galar (2017) maintenance and AM[1] still face a number of managerial, technical and methodological challenges today. While recent years have seen increasing awareness of the importance of good AM, a substantial amount of research effort and cumulative recognition of what it entails; industry is still struggling to harness the full benefit thereof as there is a lot of wasted effort.

Maintenance costs have continued rising, both in absolute terms and as a proportion of the total business expenditure. In 30 years, it went from being an insignificant contribution to a major cost priority (Arunraj and Maiti 2007; Moubray 1997). Accounting for a rapidly increasing share of the operating costs (Hipkin and De Cock, 2000), maintenance is now the second highest, or even highest cost element in some industries (Arunraj and Maiti 2007; Moubray 1997). In fact, according to Mobley (2002), it can represent up to 60% of the total operating

[1] While it is generally recognised that AM goes beyond maintenance management, maintenance remains a fundamental part of AM that is critical to an organisation's long-term profitability and survival (Kans and Galar, 2017)

costs and up to 70% according to Arunraj and Maiti (2007). In the USA alone, this amounted to an annual maintenance and AM expenditure of $200 billion in 2002 rising to $1.2 trillion in 2005 (Mobley 2002; Penrose 2008). The magnitude of this expenditure coupled with industry's growing dependence on assets and an increasingly competitive business environment has made maintenance a matter of organisational survival rather than merely a cost consideration (Kans and Galar 2017; Sharma and Yadava 2011).

However, an alarming proportion of both the bulk and growth of this expenditure is reportedly due to increasingly ineffective spending indicating a poor return on investment. It is estimated that in 2002 up to a third of all maintenance costs (over $60 billion in the USA), growing to just under two thirds ($750 billion) in 2005, were wasted as a result of poor asset maintenance and management decisions (Mobley 2002; Penrose 2008). This trend is confirmed by Carstens (2012) who states that, despite industry realising the importance of good asset management, many organisations still knowingly perform ineffective maintenance tasks; and Sharma and Yadava (2011) who states that the gap between industry and literature remains large. It is clear then that the asset management industry has great room for improvement with Penrose (2008) calling it the *"single largest business improvement opportunity of the 21st Century"*.

A significant amount of research effort has gone into furthering the field of asset management in the past few years leading to the publication of several resources by various organisations to consolidate the available expertise, identify best practices and to guide and improve the implementation thereof (IAM, 2014). Some examples of these include PAS 55, ISO 55000, the AM Anatomy and the AM Landscape (BSI 2008; ISO 2014; GFMAM 2014; IAM 2014). While the background, scope and focus of these resources are not exactly the same, the underlying principles of their content is generally well-aligned and useful for different purposes (IAM, 2014).

One of the principles agreed upon by these resources, is the importance of evidence based, data-driven AM. They recognise information as an asset in itself meaning it is not only a necessary asset-management tool but also a value-adding entity falling within the scope of assets to be managed (BSI 2008; ISO 2014; GFMAM

2014; IAM 2014). Even PAS 55, which is focussed on the management of physical assets (PAM), recognises that this is inextricably linked, and directly impacted by the management of information.

This means that like any other AM activity, data management is the responsibility of top-management, but requires stakeholder buy-in from all organisational levels. It forms part of the critically important *"line-of-sight"* between the high-level, organisational objectives and the ground-level, daily activities. Such bi-directional alignment requires not only the top-down integration of strategic direction into everyday activities; but also the bottom-up data feedback that provides input to all levels of AM (development, implementation and optimisation) to ensure information-driven decision making that is rooted in fact-based realities. (BSI 2008; ISO 2014; GFMAM 2014; IAM 2014). Furthermore, like any other asset, data must be considered from a life-cycle perspective that *"starts with conception of the need for the asset through to its disposal"* (ISO, 2014) requiring continuous maintenance, monitoring and improvement to realise value from it.

According to these resources, the availability of accurate, up-to-date asset information is a key enabler of both the strategic and operational AM activities and is, in fact, imperative for successful AM overall. This is emphasised throughout these documents and specifically in Section 4.4.6 in PAS 55 (under AM enablers and controls), Section 7.5 in ISO 55001/2 (under AM support elements), Section 5.4 in the AM Anatomy (fourth subject group) and Section 6 in the AM Landscape (subjects 22-25).

While the importance of evidence based, data-driven AM is undisputed by these and other resources; according to Mobley (2002), one of the leading causes of ineffective AM remains a lack of factual data to support business decisions. In a case-study of four leading manufacturing organisations, Hipkin and De Cock (2000) identifies the lack of historical data and insufficient time to complete analyses as two of the major barriers for effective AM. From a literature survey of the biggest challenges in AM, Marquez (2007) confirms both these issues further stating that there is a big disjoint between what practitioners think they should do (as specified by AM resources) and what they actually do. They report that managers are preoccupied by day-to-day operations and reactive problem-solving with little time left for strategic data analyses and focussed improvement (Marquez, 2007). Goble and Siebert (2008) further state that poor information management practices, and particularly issues with data quality, prevent effective utilisation of the data that *is* available. More recently, all of Braaksma and Veldman (2013), Baglee *et al.* (2015), Woodall *et al.* (2015), Reeve (2016), Karim *et al.* (2016), Kans and Galar (2017), Zschech (2018) and many more, have confirmed that a lack of data, data quality and various other data management inadequacies in areas such as collection, analysis and IT competence are major issues in AM.

Zschech (2018) creates a taxonomy of the recurring data analytics problems in AM grouping them according to the type of data going into the analysis, the analytical techniques and the source of the data. These are discussed in Sections 2.2, 2.3 and 2.4 below.

## 2.2 Data Types

Zschech (2018) identifies two main categories of data that is relevant to AM: condition monitoring (CM) and event data. CM data relates to the health condition of assets as determined by physical measurements such as vibration, temperature or pressure readings. Event data, also called failure data, relates to failure events or other asset-related incidents and non-conformities (including near misses and false alarms) as well as the subsequent actions taken (such as repair, replace or configure) (Zschech 2018; Samuel *et al.* 2006; ISO 2014).

Two additional, more general categories of data that applies to both event and CM data, are metadata and business-data (Zschech, 2018). Metadata is typically defined as data about data to provide context to either a specific data entry or the data collection system. Metadata can be *descriptive* (used for discovery and identification of a particular data entry, e.g. machine type, location, manufacturer); *structural* (describes the organisation of the data system; the required syntax and permissible range of values including lists of identifiers, their definitions and the relationships between them e.g. measurement unit, precision, list of machine types); or *administrative* (used to facilitate information management, e.g. author, date and time created, access permissions) (Aljumaili 2016; Zschech

2018). Business data describes the environmental context ranging from legal requirements to market projections to production plans and scheduling information including all revenue, profit, cost and resource information (Zschech, 2018).

The focus of this study is on event data and particularly that of failure events (hereinafter called failure data). According to Zschech (2018), this data is frequently treated as having secondary importance and rarely used beyond an operational record-keeping function. They cite an erroneous belief that CM data is sufficient, despite several authors emphasising the importance of failure data for effective AM (Zschech 2018; Syeda *et al.* 2018; Karim *et al.* 2016; Arunraj and Maiti 2007).

Failure data is imperative for monitoring progress, establishing risk and guiding further improvements; all of which are important concepts in AM. Failure is an unavoidable reality for even moderately complex asset systems, and while unfortunate, these events do provide a valuable opportunity to improve this by identifying areas of concern (Syeda *et al.*, 2018). It provides information on not only the performance of the assets, but also of the asset management, which is necessary to guide further improvement efforts and to justify maintenance programs from both a cost and risk perspective (Hipkin and De Cock, 2000). Furthermore, if these events are not investigated, the risk of failures will remain unchanged and are likely to reoccur (Syeda *et al.*, 2018).

This is confirmed by ISO 55000 which states that all *"asset related incidents... should be investigated and reviewed to see if any improvements are needed and to prevent their reoccurrence and mitigate their effects"* (ISO 55000: 2.5.3). In fact, the collection and analysis of historical failure data is explicitly mentioned as part of the *Information Requirements* needed to support AM (ISO 55000: 7.5) as it is needed for activities such as determining the AM objectives (ISO 55000: 6.2), risk assessment and management (ISO 55000: 6 and 8), performance evaluation (ISO 55000: 9) and continuous improvement (ISO 55000: 10) (ISO, 2014).

In recent years, widespread recognition of the importance of failure data has led to the majority of companies collecting at least some form of failure data (Baglee *et al.*, 2015). The ongoing industrial digitisation has made data collection easy. Large-scale computerisation and rapid advances in information technology has led to a considerable increase in the amount and variety of AM data available (Zschech 2018; Baglee *et al.* 2015; Kans and Galar 2017). However, increasing the amount of data collected does not necessarily translate to an increase in relevant information (Zschech, 2018). The challenge is not to collect as much data as possible since too much data can even impede the ability to extract useful information with the real value lying in the insights gained from the modelling and analysis thereof (Karim *et al.* 2016; Kans and Galar 2017; Baglee *et al.* 2015).

## 2.3 Analytical Techniques

Like Zschech's (2018) distinction between data types, Welte and Wang (2014) distinguish between failure-time and degradation models based on failure and CM data respectively. Failure-time models are concerned with modelling and predicting the time-period associated with failures which is typically modelled as a stochastic process (Marquez, 2007). An important aspect of stochastic models is that it involves uncertainty, namely quantities that cannot be predicted exactly, only estimated probabilistically (Welte and Wang, 2014). This is due to the inherent uncertainty associated with the deterioration process, ambiguity regarding future operation of the machine, and finally, errors associated with the analysis/modelling methods being applied (Sikorska and Hodkiewicz, 2011). In other words, while it cannot be predicted when failures will occur exactly, their probability of occurring at any given moment or within a certain interval can be estimated with reasonable accuracy using probability theory and statistical methods (Marquez 2007; Welte and Wang 2014).

Degradation models are concerned with the changing technical condition of an asset which is represented by some degradation variable over time (or usage) with failure assumed when this variable crosses some threshold (Welte and Wang, 2014). As the focus here is on failure data, degradation models are not considered further.

One of the first challenges faced by analysts is deciding how to categorise large amounts of failure data in some meaningful manner (Syeda *et al.*, 2018). Accurate analyses require a substantial amount of data (Sikorska and Hodkiewicz, 2011) and since not all failures occur with the same frequency[2], Samuel *et al.* (2006) recommend grouping them on the basis of engineering judgement to get more statistically relevant samples and evaluating these clusters together. This is supported by PAS 55 which recommends grouping failures into categories of similar events to facilitate trends analysis (BSI, 2008).

A common way of doing this is failure modes, namely categorising events according to the modes, or manner by which asset failure can occur (Braaksma and Veldman 2013; Dictionary 2018; Rudov-Clark and Stecki 2009). In fact, according to Samuel *et al.* (2006), this grouping forms part of the first step in any analysis identifying two generic steps: 1) *Failure mode identification and data preparation*; and 2) *Statistical analysis and interpretation of results.*

The first step (data preparation and particularly the classification of events into failure modes) is the focus of this study and the objective of the empirical analysis discussed in Chapters 6-8. For a discussion on the different modelling approaches the interested reader can refer to Welte and Wang (2014), Sikorska and Hodkiewicz (2011) and Marquez (2007).

Failure modes can be defined at any level of abstraction (system, subsystem, component or even material level) depending on the needs of the particular analysis. Therefore, the first step comprises data collection in the required format and the failure mode categorisation according to the selected level of abstraction (Samuel *et al.*, 2006). For the empirical analysis discussed in Chapters 6-8, failure modes are defined at the sub-assembly level (e.g. mechanical, electrical and hydraulic failure). This categorisation is supported by Devaney *et al.* (2005) who note that while highly unique, complex assets will have different components; they share high-level systems and subsystems (such as hydraulics, electronics and pneumatics) which share machine-independent characteristics with common parts, loads, functions and interconnections (e.g. all hydraulic systems have hoses which can burst). Considering them at this level provides more generically applicable results and data samples that can be shared across various domains and machinery

[2] From a case-study evaluating aero-engine failures, Samuel *et al.* (2006) reports that a small proportion of failure types accounted for more than 80% of the observed failures with the majority occurring only rarely.

(Devaney *et al.*, 2005). Most importantly however, this was the indicated preference of the data provider.

The second step involves the modelling and prediction of failure events as well as the subsequent decision making based on the interpretation of the results (Welte and Wang 2014; Samuel *et al.* 2006). Failure modelling and analysis contains a large number of methods. While the level of detail, methodology, assumptions, focus, limitations etc. of these methods may differ; most concern the same basic categories of information, (called entities of interest (EOI) by Syeda *et al.* (2018)) such as the failure time, time to failure, the time between failures or the number of failures (at any given moment or within a time, or usage, interval) (Welte and Wang 2014; Goble and Siebert 2008; Marquez 2007). These can be obtained from a variety of sources including operational field data, public databanks and domain expertise (discussed below).

The first step is a prerequisite to the second. Models and subsequent decisions are only as good as the data they are based on and as in the field of computer science, garbage in gives garbage out (Samuel *et al.* 2006; Aljumaili 2016; Woodall *et al.* 2015; Goble and Siebert 2008). Several authors confirm the importance of this step, with Kenny *et al.* (2017) stating that it is *"of significant importance to understand a device's high priority failure modes and prepare for them."* (Kenny *et al.*, 2017). However, Samuel *et al.* (2006) report that it is not uncommon to observe situations where no attention is given to the first step and that, compared to literature about statistical modelling and analysis, there is much less about failure mode identification and data preparation.

This is confirmed by Sikorska and Hodkiewicz (2011) and Marquez (2007) who state that while a substantial amount of research is available on the topic of failure modelling and analysis, much of it is of mathematical/academic interest only neglecting practical considerations such as the data, expertise and computational infrastructure requirements. In fact, according to Sikorska and Hodkiewicz (2011) there is only limited evidence of truly successful implementations in industry. However, they and other authors emphasise that this is not as a result of problems with the various tools, but rather due to poor understanding of the limitations of these tools (Sikorska and Hodkiewicz 2011; Goble and Siebert 2008; IAM 2014).

All models are subject to various assumptions and approximations by design; some mathematical and some practical concerning implementation issues such as the amount, type and quality of data required (Sikorska and Hodkiewicz 2011; Marquez 2007). Variations in these and other factors such as the maintenance capabilities of the site, their definition of failure, failure recognition method, failure data recording and collection policy (including the competence of those responsible for planning, collecting and analysing data) and assumptions used to calculate EOI, such as the number of operating hours and number of failures, can cause order of magnitude differences in the results and it is imperative to completely understand the methods used to define, collect and analyse failure data before using it (Goble and Siebert, 2008).

## 2.4 Data Sources

The data required by these analyses can be obtained through both quantitative and qualitative means. Quantitative methods use empirical observations of failure obtained from either the plant's actual performance data (historical field data/records) or from the databanks published by original equipment manufacturers (OEM) to determine the various EOI such as the number of failures. In qualitative methods subject matter experts (SME) estimate these using a variety of polling, interview, and questionnaire techniques. (Welte and Wang 2014; Marquez 2007; Arunraj and Maiti 2007)

Manufacturer data consist of the failure rates and other reliability information published by OEMs based on either laboratory tests or field warranty and return data. Neither of these are an accurate reflection of the true operating environment under consideration and may result in optimistic expectations (Hameed *et al.* 2014; Goble and Siebert 2008; Samuel *et al.* 2006). Studies have shown that asset reliability depends on a wide range of environmental and operational factors which are difficult to simulate in a laboratory environment and are not typically specified in public databanks. The data provided by these sources are average values at best and idealistic values at worst (Marquez, 2007). Manufacturer data further neglects various practical considerations such as the business-related consequences of failure, regulatory requirements and the availability of resources (Marquez, 2007).

Laboratory tests typically assume ideal operating conditions that are unlikely to be a reflection of the real world where installation or operation errors (such as poor calibration, insufficient lubrication, unexpected loads) and environmental factors (such as ambient temperature or electrical surges) are likely to cause worse, or at least unexpected, results (Samuel *et al.* 2006; Kenny *et al.* 2017).

While return and warranty records provide a better indication of the real-world operating conditions, it is still an average measure and not specific to each consumer (Hameed *et al.*, 2014). Furthermore, according to Goble and Siebert (2008), this data is optimistically skewed due to the limited information actually available to an OEM. Operational hours, estimated from the shipping and return dates, are typically *over-estimated* due to unrealistic time-from-shipping-to-usage expectations. Failure counts, on the other hand, are typically *under-estimated* due to the unrealistic assumption that all field failures are reported (Goble and Siebert (2008) report on a survey finding that only 10% of failures are reported to manufacturers). Not only does this skew the type of failures recorded, but combined with the overestimated operational hours, also leads to dangerously optimistic failure rates being published (Goble and Siebert 2008; Samuel *et al.* 2006).

Field data is typically connected to an organisation's work-order system (Goble and Siebert, 2008) which stores all the job-related information for every maintenance activity/event (Uz-Zaman *et al.*, 2015) in a Computerised Maintenance Management System (CMMS) (Aljumaili, 2016). These records are primarily focussed on the day-to-day operational issues such as scheduling and communication after which they are archived for operational record-keeping purposes (such as settling disputes, tracking costs and documenting labour) (Woodall *et al.* 2015; Palmer 2006). While they explicitly describe every problem, repair, adjustment and alteration made to the assets of an organisation, they implicitly contain much higher-level information that can be used to construct life-cycle models, identify repair and failure trends, optimise maintenance schedules, and ultimately, support long-term strategic decision-making (Devaney *et al.* 2005; Palmer 2006).

Field data has many advantages and has been described as the ultimate source of failure/reliability information (Goble and Siebert 2008; Samuel *et al.* 2006). It reflects the unique operating conditions of each organisation providing insight to the specific problems, risks and opportunities faced by that organisation and is therefore expected to yield the most accurate results (Kenny *et al.*, 2017). Furthermore, because this data is in direct control of the organisation in question, they are able to not only know the limitations of their data (in terms of quality, bias etc.) but can also address these limitations and tailor the data to their specific needs (what to collect, how to collect, level of precision etc.) (Goble and Siebert 2008; Samuel *et al.* 2006; Kenny *et al.* 2017). Almost all organisations have this type of data available although the quality may vary from plant to plant (Goble and Siebert 2008; Mukherjee and Chakraborty 2007; Uz-Zaman *et al.* 2015).

However, these datasets often suffer from several data quality issues restricting effective utilisation of the available information (Rajpathak *et al.*, 2012). Records are typically entered directly into a database from the operating environment by maintenance personnel via fairly limited devices under sometimes severe time constraints (Devaney *et al.*, 2005). Poor understanding of how the data will be used or the consequences of low data quality lead to maintenance personnel treating data collection a secondary task with low importance. This is especially prevalent in low-maturity organisations which are characterised by departmentally fragmented silos and poor line of sight (Woodall *et al.* 2015; Braaksma and Veldman 2013). Because the data is not collected by the end-user, it is often initiated without clear purpose of use (collecting for the sake of collecting) and in the wrong format (easy to record, but difficult to analyse) (Reeve 2016; Woodall *et al.* 2015; Devaney *et al.* 2005).

Operation and maintenance staff will typically store failure data in free-format, natural language text which is not conducive to computerised analytics (Reeve 2016; Uz-Zaman *et al.* 2015; McKenzie *et al.* 2010). Extracting meaning from such data must be done manually which is incredibly tedious, time-consuming and cumbersome work (Chen and Nayak, 2007).

By contrast, qualitative data obtained from subject matter experts (SME), do not require expensive data collection or processing (Arunraj and Maiti, 2007). Domain expertise can either be incorporated directly into AM decisions (e.g. by selecting maintenance intervals) or indirectly by providing the inputs for the various models, tools and techniques such as a conceptual FMEA[3] performed using a qualitative risk assessment. While qualitative data is more quickly and easily available requiring much fewer resources; these benefits come at the cost of precision. It can yield satisfactory results if collected with due diligence, but because it relies on the opinions, experience and intuition of individuals it is inherently subjective and difficult to enforce any degree of consistency or uniformity (Marquez 2007; Arunraj and Maiti 2007).

Domain expertise is a valuable asset in any organisation but suffers from poor transferability and might not be held by those responsible for decision making (Woodall *et al.*, 2015). It is only as good as the knowledge and experience of those involved and, because the information is contained in individuals, it is subject to expert availability which may cause organisations to become dependent on staff members (who might leave) (Braaksma and Veldman 2013; Sikorska and Hodkiewicz 2011). Furthermore, human intuition is subject to gross oversight and there is often (if not always) a big difference between perceived and actual risk (Tversky and Kahneman 1973; Plous 1992).

In any given situation, people are bombarded with massive amounts of information that the brain cannot possibly cope with. For this reason, the brain develops various heuristic principles by which to simplify complex information management (Tversky and Kahneman, 1973). Many of these heuristics have been studied and documented by psychologists. Although useful (and necessary) for quick decision making and approximations, they can sometimes lead to severe and systematic errors (Hobbs and Reason, 2003). One of the most relevant brain short-cuts is known as the *"availability heuristic"* (Plous, 1992). It refers to people assessing the probability of an event (or the frequency of its occurrence) by the ease with which instances of occurrence can be recalled. The brain will typically be able to recall recent events much easier than older events leading to a time-dependent

[3] Failure Mode and Effects Analysis (FMEA) is a common AM tool used to prioritise assets, actions and spending according to the risk computed from some combination of the probability of failure and its consequences. In a conceptual FMEA, these are estimated by domain experts, but the probability and impact rankings can also be quantified using real frequency and monetary/production data. (Kenny *et al.* 2017; Arunraj and Maiti 2007; Hameed *et al.* 2014)

bias that will vary according to when a person is polled (Tversky and Kahneman, 1973). Moreover, people tend to recall dramatic events much easier than less severe ones, regardless of the statistical relevance of the incident. This phenomenon, called *"risk telescoping"*, leads to underestimating low-impact, common risks and overestimating high-impact but rare risks (Science, 2016). These are just some of the more common inaccuracies that can occur when making intuitive judgements rather than using statistical analyses of historical data[4].

Several authors have investigated this mismatch finding substantial differences in all but the simplest of systems. Kenny *et al.* (2017) evaluates the reliability of domain expertise by comparing the perceived risks (obtained from a conceptual FMEA) to the actual risks (obtained from historical failure logs) for a Squid 6 Series Wave Energy Converter System. Domain experts tasked with completing a conceptual FMEA assigned probability, consequence and detection rankings to each failure mode to determine their respective risk priority levels. To ensure a consistent comparison, the authors created a failure recording template identical to this FMEA which was retrospectively populated using information found in the field records. Samuels reports on a similar case study comparing a conceptual FMEA with historical data for a TFE 731 aero engine.

Both studies found substantial differences between the perceived and actual risks presented by these two assessments. Kenny *et al.* (2017) reports numerous variations of either one or both of the frequency and impact rankings being over or underestimated. Whilst not exploring the topic themselves, their results further seem to provide anecdotal evidence of the availability heuristic. In particular they found that high impact failures tended to overshadow those of lower impact leading to a disproportionate focus on rare, but high consequence events (risk telescoping). Moreover, where low impact, high frequency failure modes were overpredicted, it supports the premise of a time-related bias as experts tended to over-predict corrosion, fatigue and wear which become increasingly significant towards the latter

[4] This is by no means a psychology paper and does not attempt to fully explore the psychological factors involved. It merely provides a brief overview of some of the relevant theories to illustrate the problems inherent in intuition-based identification, and analysis of, failure modes. The interested reader can refer to Hobbs and Reason (2003) who provides a more in-depth study into the reasons for maintenance error.

(and more recent) stages of asset life (Kenny *et al.*, 2017). While (Samuel *et al.*, 2006) provides less detail of the individual variations, they support the overall results finding only 20% agreement between the risks perceived by domain experts and that observed in the field data.

Regardless of the reason for these discrepancies, the fact remains that either over or underestimating the risk of various failure modes can have adverse business consequences due to the increased costs of non-essential maintenance efforts in overestimated areas; and the increased costs of safety and production issues in underestimated areas hampering the efforts to reduce or eliminate risk effectively (Braaksma *et al.* 2011; Arunraj and Maiti 2007; Samuel *et al.* 2006; Kenny *et al.* 2017). In fact, from a case-study evaluating the effectiveness of industrial applications of FMEA, Braaksma and Veldman (2013) found limited evidence that it actually supported consistent decision-making or continuous improvement due primarily to the sole reliance on domain expertise rather than field data to complete them.

Both Samuel *et al.* (2006) and Kenny *et al.* (2017) report that such qualitative risk assessments are inadequate for all but the simplest of asset systems as it becomes increasingly difficult to conceptualise complex, real world systems. They relent that in the absence of field data (such as the design phase of a new asset prototype or when field data has simply not been collected) domain expertise can provide a valuable estimate of asset reliability; but emphasise the importance of verifying this with field data as soon as it becomes available. This is supported by Arunraj and Maiti (2007) who recommends that qualitative risk assessments should only be used when risks are small and well-understood or when no field data is available.

However, even when field data is readily available, according to Baglee *et al.* (2015), Braaksma and Veldman (2013) and Palmer (2006) it remains a challenging task to convince managers to trust this over their own intuition. While Goble and Siebert (2008) state that most of the field data required by failure analytics is already being collected by organisations, according to Reeve (2016) as many as 70% do not perform even basic analyses on their field data and remain almost wholly reliant on domain expertise. Even more concerning, they state that this percentage has not changed in many decades (Reeve, 2016). Several authors confirm that a sole reliance on expert judgement is a major problem in AM (Sikorska and Hodkiewicz 2011; Syeda *et al.* 2018; Reeve 2016; Baglee *et al.* 2015) with Braaksma and Veldman (2013) citing a widespread belief that field data is not of adequate quality to be useful.

This is not an entirely incorrect assumption, because while Kenny *et al.* (2017) agree that operational data is superior to predicted data, they also note the significant challenges involved with utilising this data retrospectively. A lot of valuable information is buried in unstructured natural language text (rather than constrained value fields) which is difficult to analyse (Syeda *et al.*, 2018). Included in this is the failure modes, which is the focus of this study, as Reeve (2016) found that most CMMS products do not explicitly record this information. The reason for this is partly historical.

Traditionally, all record keeping was performed through paper-based systems. Such hard-copy records require manual, log-by-log analysis (McKenzie *et al.*, 2010). This meant that unstructured textual descriptions posed no significant analysis disadvantage as, unlike computers, humans are not able to process structured, numeric inputs much faster than free text. This made text, which offers the advantage of greater expressivity and the convenience of translating directly from human thought and communication, a natural choice. Upon the integration of computers into everyday life, (most notably through CMMS) most organisations transferred to electronic data collection systems, but many retained the same basic format, including the free text fields (McKenzie *et al.*, 2010). This means that despite being stored in an electronic format, only the structured fields can benefit from faster computer processing while the text fields still require time-consuming and labour-intensive manual processing which few can afford (McKenzie *et al.* 2010; Chen and Nayak 2007; Rajpathak and De 2016; Devaney *et al.* 2005).

For CM data consisting of physical measurements, numeric inputs were a natural choice and the analysis thereof is a well-researched field with many successful applications such as vibration-based condition monitoring (Rajpathak and De 2016; Wang *et al.* 2017). However, for event data (which is the focus of this study) the structured fields are mostly operational identifiers such as date and time reported or location ID; with the more valuable strategic information (such as failure modes) implicit in the unstructured text fields. This is due to the heterogeneous nature of every failure event. Capturing the distinct details of every incident is a challenging task greatly aided by the expressivity of natural language text (Chen and Nayak 2007; Syeda *et al.* 2018).

The problem of unstructured text data, and the broader issue of ineffective information management, is not isolated to the field of AM. It has been estimated that up to 80% of all organizational data is in an unstructured, text format that is not amenable to traditional analytics (Singh and Raghuvanshi 2012, Kobayashi *et al.* 2018; Gupta and Lehal 2009; Tan 1999; Gutierrez 2015). Moreover, according to the Electronic Commerce Code Management Association (ECCMA) poor data management adds up to 20% to companies' direct and indirect costs (ECCMA, 2019). The data creating capabilities of many industries have far surpassed the ability to handle it with many companies hoarding vast amounts of data they will never use (Russom, 2011). Not only do they not benefit from this data, but it has been recognised that excessive amounts of information may even have a negative organisational impact because it distracts from the main issues, delays decision making and can lead to wrong conclusions if incorrectly utilised (Wuest *et al.*, 2016). In fact, this overwhelming flood of information, or *"Big Data"* as it is called, has led to 30% of companies viewing data as a problem rather than the opportunity it could be according to the Transforming Data With Intelligence (TDWI) Best Practices Report (Russom, 2011).

## 2.5 Big Data

Technological advancement has led to an exponential increase in the amount of data generated worldwide with faster computers generating an ever-increasing flood of information at ever-increasing speeds making conventional data storage, processing and analytics increasingly ineffective (Hurwitz *et al.*, 2013). Big Data (BD) is characterised by the fact that data is coming from a wider variety of sources, and in a wider variety of formats, than ever before (Russom 2011; Kans and Galar 2017; Karim *et al.* 2016). A lot of it is also being generated by outside sources and used for different purposes than it was originally collected for. Because the analyst was not in control of (or even involved in) the data collection, there are frequently concerns regarding the quality, credibility and suitability of such data (Sathi 2012; Witten *et al.* 2011). However, along with these challenges it is also recognised that there is significant potential in BD (Kans and Galar, 2017).

Companies can now know so much more about their clients, their suppliers, their business environment, their own production and even their employees. Having such large amounts of data available create many opportunities for businesses that have the potential to yield big returns for relatively small investments. Data-driven decisions are necessarily better decisions because managers can make use of evidence in previously intuition-based environments. *"You can't manage what you don't measure"* is a popular business maxim used to demonstrate the importance of utilising the available data (McAfee and Brynjolfsson, 2012).

There are no threshold values for when data becomes Big Data. Rather, it is any data that cannot be processed using traditional means and is typically characterised by five V's: Volume, Velocity, Variety, Veracity and Value (Kans and Galar 2017; Karim *et al.* 2016; Syeda *et al.* 2018; Baglee *et al.* 2015).

Volume and velocity refer to the amount and speed of data generation. What makes the velocity aspect of Big Data particularly challenging is that often, for the data to be of any use, the analysis must also happen in real time (e.g. checking credit card transactions for fraud detection) (Russom, 2011). Variety concerns the difficulties involved in treating the broad range of data formats that contain valuable information but do not come in structured, easy-to-analyse data sets; such as images, video, voice recordings and of course text (Syeda *et al.* 2018, Russom 2011). Veracity concerns the quality and credibility of the data which can be negatively affected by bias, noise, duplication and error; be it as a result of the source, processing, type or format of the data (Sathi, 2012).

The final characteristic of Big Data is that it must have value. It is imperative to understand the costs and benefits of collecting and analysing the data before embarking on expensive BDA initiatives to ensure tangible business benefits. This concerns the potential usefulness of the analytical outcome as simply having large quantities of data has no real value in itself (Karim *et al.* 2016; Baglee *et al.* 2015).

In the analysis of field failure data, variety and veracity are especially big issues (Galar *et al.*, 2015). As discussed previously, a major challenge preventing the analysis of field failure data is the descriptive text fields which are not amenable to traditional analytics due to its unstructured nature (variety) (Kans and Galar 2017; Karim *et al.* 2016; Galar *et al.* 2015). This ties in with the problem of veracity, as text is a naturally noisy data source due to the inherent inconsistencies and variation in human language. Different people will use different words to describe the same failure and even a single person is highly unlikely to be perfectly consistent when describing different occurrences of the same event (Tumer *et al.*, 2003). Moreover, because the data is typically not collected by the end-user and is recorded in fairly haphazard ways with little to no understanding of why or how the data will be used; these datasets typically suffer from several additional data quality issues making veracity a particularly pertinent concern (Reeve 2016; Woodall *et al.* 2015; Devaney *et al.* 2005).

Several authors have also confirmed that the analysis of field failure data, and maintenance work order records in particular, produce a number of benefits for an organisation; all of which can be summarised as saving money (Goble and Siebert, 2008). More specially, the information contained in these records can be used to create reliability models (Rajpathak and De 2016; Mukherjee and Chakraborty 2007), identify best-practice repairs (Rajpathak, 2013), accurately predict maintenance budgets (Edwards *et al.*, 2008), reduce downtime and prevent failures (Devaney *et al.*, 2005), improve inventory and spare part management (Rajpathak *et al.*, 2012) and any number of other activities that enhance both the strategic and operational decision-making processes (Rajpathak and Chougule, 2011).

While neither the volume nor the velocity is of particular concern for maintenance datasets (which are small compared to many BD problems), it falls in the realm of BD because it is not amenable to traditional analytics. Accordingly, several authors have suggested looking at the analytical tools and techniques which have had success in this field, namely Big Data Analytics (Syeda *et al.* 2018; Galar *et al.* 2015; Zschech 2018).

### 2.5.1 Big Data Analytics

Big Data Analytics strives to extract useful information from large quantities of raw data and can therefore be described as a Data Mining (DM) problem; the analogy being that, like the extraction of valuable minerals from Earth, large amounts of low-value raw material must be processed and sifted to yield a much smaller quantity of some precious substance (Alpaydin 2010; Witten *et al.* 2011). However, to differentiate from those using DM to describe only a single step in the knowledge discovery process, Sharma (2008) recommends using the term Knowledge Discovery and Data Mining (KDDM) to emphasise the inclusion of the entire process; from data storage, access, and processing to the analysis and interpretation of the results.

The difficulty in analysing natural language text data comes from its unstructured (and therefore unpredictable) nature (Hurwitz *et al.*, 2013). Computers make use of logic-based algorithms and have no "understanding" of language in the human sense. They are only capable of binary comparisons, namely exact-match or not; there is nothing in-between even if the only difference is as small as a single capitalisation. This makes dealing with the ambiguities of everyday language (such as homonyms, synonyms and context-dependant definitions) nearly impossible to deal with using traditional analytics (Hotho *et al.*, 2005).

KDDM is an interdisciplinary approach that combines methodologies and techniques from various research fields including statistics, pattern recognition, mathematical modelling, data visualisation, optimisation and high-performance computing (Zschech 2018; Maimon and Rokach 2010). Particularly relevant here is text mining (TM); the automated, or semi-automated (computer-assisted) extraction of useful knowledge from text data (Wachsmuth, 2015). While there is some contention in literature as to the exact definition and scope of text mining, it is generally accepted that it is a multidisciplinary field that draws heavily from the fields of machine learning (ML) and natural language processing (NLP) with some seeing it as a branch or sub-field of data mining and others as a distinct but correlated, sibling field (Dhanrajani and Gosh 2008; Witten *et al.* 2011).

For this paper, text mining is used to mean text data mining with data mining defined as the exploration and analysis of large quantities of data by automatic or semi-automatic means with the purpose of obtaining useful knowledge (paraphrased from the original definition by Berry and Linoff (1997) cited in Bastos *et al.* (2014)). Machine learning can be seen as the technical basis of data (or text) mining and refers to the area of artificial intelligence that allows computers to "learn" from data rather than being explicitly programmed (Bastos *et al.* 2014; Witten *et al.* 2011). Natural language processing, which is a sub-field of artificial intelligence and computational linguistics, is the broad term for the integration of human language and computers that is concerned with converting natural language to machine-readable, structured data (Dhanrajani and Gosh, 2008). This means that NLP can be both a text mining tool and a text mining outcome.

The analysis of text-based maintenance records is a typical text mining problem according to Chen and Nayak (2007); but they (and other authors) acknowledge several domain specific challenges not typically addressed in text-mining literature. This includes short document length, a lack of domain specific benchmark datasets and the use of non-standard English (such as domain specific jargon, short-hand notation and a proliferation of spelling and grammar errors) (McKenzie *et al.* 2010; Devaney *et al.* 2005; Edwards *et al.* 2008). Of course, this is not the only solution to address the AM data needs.

## 2.6 Alternative and Ideal Solution

The collection of free text records is widely recommended against in the general literature where several authors recommend using a consistent, predefined structure and format to collect failure data explicitly; preferably by selecting coded inputs from a short list of predefined information entities (Reeve 2016; Palmer 2006; Marquez 2007; Woodall *et al.* 2015; Rajpathak *et al.* 2012; Braaksma and Veldman 2013; Goble and Siebert 2008) or at the very least standardising the terms and definitions used in text descriptions (Chen and Nayak 2007; Rudov-Clark and Stecki 2009; Tumer *et al.* 2003).

Free text also goes against the general principles of the AM resources which require determining the information needs prior to collection and ensuring its suitability for all objectives (ISO, 2014), and more specifically, that failure data must be captured in a manner that enables subsequent analysis (GFMAM 2014; BSI 2008). The ideal solution would therefore be to prevent the collection of such data by improving the data collection and information management practices of an organisation (Kenny *et al.* 2017; Braaksma and Veldman 2013; Reeve 2016; Rajpathak *et al.* 2012).

There are several resources available, both within AM domain and beyond, to guide organisations with the implementation and management of information systems that will allow them to properly collect and utilise their failure data. Within the AM domain, Braaksma and Veldman (2013) recommend the use of asset information standards which is also one of the 39 AM subjects described in the GFMAM Landscape. Asset information standards are *"data models for unified description of information relating to assets or products"* (Braaksma *et al.*, 2011). They specify a consistent structure and format for the collection, categorisation and storage of asset information (including failure data) (GFMAM, 2014).

Beyond the AM domain, the more generic ISO 8000 data quality series is a valuable resource that is specifically recommended by the IAM Anatomy (IAM, 2014). This standard defines the principles of data quality; the data characteristics that determine its quality; and the requirements for achieving, measuring and improving data quality (Benson 2008; ISO 2011). The collection of unstructured, natural language failure data is in direct contravention of one of the leading principles of data quality identified by ISO 8000: that data must be *"fit for purpose i.e. the decision it is used in"*. ISO 8000 directly addresses this inadequacy by suggesting the use of semantic encoding: *"the technique of replacing natural language terms in a message with identifiers that reference data dictionary entries"* (ISO 2011; Benson 2008).

The data dictionary consists of a comprehensive list of entries containing, at a minimum, an unambiguous identifier, a term, and a definition of said term. These entries are allocated according to the data specifications which are the rules used to describe items belonging to a particular class. This standard does not completely discount the value of descriptive natural language text and recognises that it does have a place within the data architecture. The natural language text is, however, moved (and limited) to the data dictionary definitions. Technicians therefore record incidents using only the codes specified by the data dictionary; the result of which can be easily evaluated using traditional data analytics[5] (ISO, 2011).

This type of structure enables fully automated, computerised processing that allows for instant search and recall, automatic sorting and analysis, and essentially enables organisations to extract meaningful information from their failure data that can be used in AM decision-making and processes.

However, there are a few drawbacks which limit the practical value of this approach. Braaksma *et al.* (2011) review the most prominent asset information standards finding fairly limited industry adoption other than in select industries such as aerospace where adoption is regulated by government. While ISO 8000 was not yet available at the time of their article, and therefore not included in their review, it faces many of the same challenges. Braaksma *et al.* (2011) groups these into standards-related and organisation related problems. Issues related to the standards themselves include the slow development of standards causing them to lag behind industry practices and technologies; instability as a result of frequent revisions (often without backwards compatibility); the complexity of standards and their proposed data models (including terminological and structural confusion); and the cost of proprietary standards which can be prohibitive for small businesses (Braaksma *et al.*, 2011). Organisational challenges refer to organisational readiness, resistance to change and the absence of a clear business case (Braaksma *et al.* 2011, Braaksma and Veldman 2013).

Organisational readiness refers to an individual organisation's capability to implement and use such standards which may be hampered by a lack of resources, competence, organisational discipline, management support and commitment (Braaksma *et al.*, 2011). A major information system overhaul (as would be required by ISO 8000 and similar documents), require significant initial investments of time, money and expertise as well as sustained motivation and discipline from all stakeholders and a continued commitment by management to provide sufficient resources (ISO

[5] These are not the only aspects of data quality addressed by ISO 8000. Other important elements identified in the standard, like data provenance, accuracy and completeness, are also imperative for achieving good data quality but semantic encoding is the most directly relevant to the subject at hand

2011; ISO 2014; Mobley 2002). A system that is not supported by management has a very small chance of succeeding (Goble and Siebert, 2008). However, in a survey by Hipkin and De Cock (2000), a lack of top management support was identified as one of the highest-ranking barriers to the implementation of new AM systems in what the authors call management interference rather than support.

Both ISO (2014) and BSI (2008) emphasise that before embarking on ambitious data collection exercises, organisations must compare the cost and complexity of doing so with the value that can be derived from it. However, it is very difficult to make a clear business-case (in terms of costs and benefits) for investments in data-management. At the time of implementation, when a large amount of resources must be committed to; the benefits of future data not yet collected are indirect, intangible and unproven while the costs are easy to calculate, substantial and immediate (Braaksma *et al.* 2011, Braaksma and Veldman 2013, Hipkin and De Cock 2000). This leads to companies prioritising short-term operational needs over long term focussed improvement processes that do not yield immediate returns (Mahmood *et al.*, 2015). Low maturity companies especially struggle with this as they are often caught up in fire-fighting immediate problems with managers unwilling or unable to invest the time and money needed to overhaul the information management system. This is confirmed in a survey by Braaksma and Veldman (2013) in which respondents reported that improving data collection practices was not a priority compared to immediate problems.

Furthermore, the skills needed to create and maintain such databases are often not held by, or cheaply available, to individuals or companies who need to store and report on data (Edwards *et al.*, 2008). It has been shown that the potential value of IT system investments is directly correlated to the level of IT competence held by an organisation; which is often lacking in maintenance organisations (Kans and Galar 2017; Braaksma and Veldman 2013). This is confirmed by Aljumaili (2016) who adds that maintenance company culture frequently does not support IT solutions. Personnel perceive it as a threat to their capabilities (Aljumaili, 2016) or part of management's underlying agenda of reducing costs and labour (Hipkin and De Cock, 2000) which can lead to active resistance, indifference or even vicious compliance (Palmer, 2006). If the entire organisation is not convinced of the usefulness of these changes, any intervention is likely to fail (Aljumaili, 2016).

Like AM, effective information management is a continuous process requiring sustained effort, resources and motivation without which an organisation can easily slip back to their previous, or a different but equally undesirable, state; minus the resources invested to get there. However, according to Hipkin and De Cock (2000), managers increasingly see their role as setting things into motion with poor follow-through in what Aljumaili (2016) calls an *"implement and forget"* attitude. Often ambitious projects are set into motion and start off well, but within a short while the enthusiasm wanes with management perpetually looking for a next *"cure all"* solution (Hipkin and De Cock 2000; Palmer 2006).

The continuous maintenance of the data dictionary (or similar structure) is especially important. If the data dictionary becomes outdated, inaccurate or incomplete (e.g. if a new failure mode occurs with no corresponding data dictionary entry) the organisation can end up with worse quality data than before as technicians will have to record closest fit (but incorrect) codes (Woodall *et al.*, 2015). Likewise, if staff are not held accountable or are unable to see the promised effects of changed data collection practices (e.g. persistence of intuitive rather than data-driven decision-making); support is unlikely to continue with staff becoming demotivated (Palmer, 2006). This can lead to knowingly recording incorrect information by simply entering any code to pass field validation checks (Woodall *et al.*, 2015).

Where before the data was merely in an unusable format (but accurate), the data will now be incorrect but easily analysed (and as such still used for decision making). Decisions made with incorrect data are worse than intuition based decisions using no data as they are accompanied by a false sense of confidence (Tversky and Kahneman, 1973).

Finally, and perhaps most importantly, this will only improve the quality of data collected in the future whereas many companies have years of unused, potentially valuable, historical data that will then go to waste. This can be especially problematic for long-term assets as Kenny *et al.* (2017) emphasises the importance of collecting failure data for the entire life-cycle of an asset, from acquisition to disposal, which can span years for some equipment. In the meantime, models have to be constructed with the available data and while present-day changes will improve future AM, it does nothing to alleviate current data needs (Kenny *et al.* 2017; Mukherjee and Chakraborty 2007).

In other words, although a complete overhaul of an organisation's information management system (or at a minimum changing the way in which data is collected) is ideal, it is not a practical solution for maintenance departments that do not have consistent managerial support and discipline. Also, even for companies that are able to completely overhaul their information management system, there is still value in being able to extract information from bad quality data collected in the past. This is confirmed by McKenzie *et al.* (2010), Rajpathak and De (2016), Mukherjee and Chakraborty (2007), Wang *et al.* (2017), Sipos *et al.* (2014), Uz-Zaman *et al.* (2015), Edwards *et al.* (2008) and Devaney *et al.* (2005), all of whom identify some form of text mining as potential solution and is the approach followed for this thesis.

This approach is in accordance with the AM standards which recognise that real-world constraints (such as budget, time or competence) may prevent otherwise "ideal" decisions (IAM 2014; Palmer 2006) and strongly recommend considering solutions that do not require additional investments but leverages existing processes and data instead (ISO 2014; Marquez 2007). Furthermore, in light of continuous improvement, both ISO (2014) and BSI (2008) emphasize the importance of actively seeking out new tools, techniques, technology and practices that relate to AM and evaluating their potential benefits to an organisation and incorporating them into the AM system if appropriate.

# Chapter 3

# Literature Review: Theoretical Overview

This chapter provides an overview of the most important concepts, theory and terminology of machine learning and natural language processing which are both extensively used for text mining applications. In so doing it addresses the third project objective, the outcome of which forms the theoretical basis for the remainder of the document.

Included in this review is instructional literature such as text books and tutorials as well as the more academic literature which is focussed on specific aspects of machine learning (both those concerned with theoretical derivations and those concerned with experimental analyses).

The primary concern of industry applications is ensuring the validity of the text mining outcomes (Kobayashi *et al.*, 2018). Accordingly, this is a major focus of this chapter and is addressed mostly from a machine learning perspective. Natural language processing mostly concerns the data preparation of text and its relevance to machine learning is discussed where appropriate.

Most of the content presented here is standard practice in the respective machine learning, text mining and natural language processing domains. They are used extensively in literature and the true origin is not always apparent. Therefore, references are only supplied for the noteworthy, non-standard or controversial elements and the interested reader is referred to any number of the good text books available on the subject such as that from Alpaydin (2010), Maimon and Rokach (2010), Witten *et al.* (2011), Aggarwal (2018), McCallum (2012) and Raschka (2015) to name a few.

While there exists a considerable amount of text mining and machine learning literature, it suffers from a lack of standardisation in terms of both methodology and terminology (Moreno-Torres *et al.*, 2012). This is partly due to being relatively new disciplines and partly due to simultaneously developing from multiple different fields with different objectives, terminologies and standard practices. The wide range of algorithms, theories and diverging opinions which are documented in literature poses an especially big problem for novice practitioners (which industry, and maintenance, practitioners are likely to be) and represents a barrier for industry adoption (Wuest *et al.*, 2016). This is further exacerbated by the inconsistent use of terminology with different terms used to describe the same concept, or even worse, using the same term to mean different things. Not only does this make it difficult to find relevant literature, but it also hampers effective knowledge sharing, fair comparisons between studies and the identification of best practices (Moreno-Torres *et al.*, 2012).

## 3.1 Machine Learning

Machine learning is a subdiscipline of both statistics and artificial intelligence that allows a computerised system to evolve (and improve its performance) when exposed to new data. The term was first used in 1959 to describe the *"field of study that gives computers the ability to learn without being explicitly programmed"* by Arthur Samuel[1] (who demonstrated this by developing a computer program that learned how to beat him at checkers) (Schuld *et al.* 2015; Samuel 1959). Since then the field has advanced significantly and has been successfully implemented in

[1] Although this definition is widely quoted in literature, the original source of the statement could not be found. Authors either cite secondary publications or falsely reference Samuel's 1959 "Some studies in machine learning using the game of checkers" paper (Schuld *et al.* 2015; Samuel 1959).

a wide variety of applications such as email spam filters, internet search engines and facial recognition software (Alpaydin, 2010).

Classic machine learning is divided into two main categories, namely supervised and unsupervised learning (Zhang *et al.*, 2015). More recently researchers have identified semi-supervised learning as a third in-between category (Witten *et al.*, 2011). The main distinction between these methods is the availability, or rather use, of labelled training data to "supervise" the learning process. Supervised methods are exclusively trained with labelled data, unsupervised methods with unlabelled data and semi-supervised methods with both (Witten *et al.*, 2011).

Labelled data consists of a sample of input-output pairs which illustrate the learning objective (Maimon and Rokach, 2010). For example, a supervised email spam-filter would be trained on a sample of emails labelled as spam or ham (not spam) respectively. These algorithms look for patterns and relationships between the input and output data provided in the training sample to learn a generalisable mapping function that can be used to predict the labels of new, unseen inputs (Maimon and Rokach, 2010).

In unsupervised learning, the algorithm receives only unlabelled training data in which the algorithm searches for an underlying patterns and structure. This previously unknown structure is the desired output, giving the user new and valuable information about the data. One of the most common unsupervised learning applications is clustering, an exploratory data mining activity that organises the data into groups on the basis of similarity (Hastie *et al.*, 2009).

The ability, or rather the requirement to learn from labelled data has both advantages and disadvantages. Labelled data allows the user to specify a concrete, measurable learning objective. This has many advantages as it enables directional learning, objective evaluation and model optimisation. By specifying both the input and the output, the practitioner is also able to guide the supervised learning process to produce useful models (such as email spam identification). In unsupervised learning there is no guarantee that the output will be useful and while it may uncover compelling new insights from the data, it can just as easily find banal, uninteresting and even spurious patterns (Witten *et al.*, 2011). Furthermore, because some of the labelled data can be used as a test case, it is very easy to evaluate and optimise supervised models. In contrast there is no direct measure of success for unsupervised learning algorithms (Hastie *et al.*, 2009).

However, just as the ability to learn from labelled data is an advantage in supervised learning, so the requirement thereof is a major disadvantage. While unlabelled data is cheap and abundant, labelled data is expensive and rare (Witten *et al.*, 2011). Labelling must usually be done by hand which is an incredibly time-consuming and sometimes prohibitively expensive task. Moreover, sometimes labelled data is simply not available. Supervised learning is also subject to mislabelling and can run the risk of being over-fitted to the provided data resulting in poor generalisability outside of the training sample (Hastie *et al.*, 2009).

Semi-supervised learning lies somewhere between supervised and unsupervised learning. It attempts to harness the advantages of both labelled and unlabelled data by using a small amount of labelled data to enhance a large amount of unlabelled data (Witten *et al.*, 2011). This may seem like the ideal compromise, but several studies have shown that while the addition of unlabelled data certainly helps in some applications, in others it does not and may even worsen the performance (Zhu and Goldberg 2009; Nowak *et al.* 2009).

The problem of assigning failure modes to maintenance records is well suited to the supervised domain and particularly its subfield of classification. Classification refers to the process of assigning input data to a *predetermined* and *finite* set of categories according to the input data characteristics. The predetermined qualifier distinguishes supervised classification from its unsupervised counterpart, clustering, which looks for the natural groupings in the data (Maimon and Rokach, 2010). Because supervised models learn by example, the user can identify, or predetermine, meaningful categories (such as failure modes) and train a classifier on inputs grouped accordingly. Unsupervised models, on the other hand, receive no such guidance and the natural groupings found may or may not be of interest to the user. The finite qualifier distinguishes classification from the related field of regression which predicts a non-finite, continuous variable (such as cost or time) (Alpaydin, 2010). Since failure modes are both a predetermined and finite set of categories, the problem area identified in Chapter 2 is best suited to classification.

Classification problems are further designated as being either binary or multiclass classification according to the number of user-defined classes in the output (Sokolova and Lapalme, 2009). In binary classification there are two classes of interest (although it can also be thought of as positive and negative examples of one class) while in in multiclass classification there are multiple (more than two) classes (Zimak, 2006).

The vast majority of ML literature is focussed on binary classification with the theory and algorithms developed for this scenario and authors simply adding statements such as *"This technique can easily be extended to the multi-class setting."* (Zimak 2006; Hoens *et al.* 2012; Rifkin *et al.* 2003). However, according to Zimak (2006), the extension to the multiclass setting is often not trivial.

The increased complexity of the multiclass output space inevitably increases the complexity of the underlying learning problem (Zimak, 2006). Multiclass classification is inherently more difficult than binary classification because the algorithm must construct a larger number of decision boundaries (Hoens *et al.* 2012; Rifkin *et al.* 2003). In binary classification the classes are complementary meaning that rejection of one class is sufficient to assign it to the other class while for multiclass classification, each class must be explicitly defined (Rifkin *et al.*, 2003). Moreover, errors can occur in the construction of any one of the many decision boundaries meaning that the opportunity for errors is much more significant in the multiclass scenario. Even if making random predictions, one can expect to be correct approximately 50% of the time on binary data, reducing to the order of 1/M on a multiclass problem with M classes (Rifkin *et al.*, 2003).

Multiclass problems can be decomposed into a set of binary problems and their predictions combined in an ensemble to make a final multiclass prediction using some type of voting strategy. This uses a divide-and-conquer strategy following the rationale that a difficult, complex problem can be solved by solving a set of easier, simpler problems (Rifkin *et al.*, 2003). There are multiple reasons to do this. The most frequent reason is that some algorithms are only suitable for binary classification and do not have a multiclass formulation (Hoens *et al.*, 2012). In this case the user has no choice. However, sometimes even when there are multiclass formulations, a binary ensemble could provide advantages. Such decompositions provide a powerful tool to transform the less studied multiclass problem into the more studied binary problem (Hoens *et al.*, 2012). Binary problems are easier to optimise and can sometimes help to distinguish between closely related classes as it focuses on one distinction at a time (rather than all at once). However, Zimak (2006) cautions that the separate optimisation of a collection of independent classifiers might not correspond to the optimisation of the overall objective and can lead to suboptimal, if not poor, multiclass results.

The most common decomposition schemes are *one-versus-rest* (frequently misnomered as one-vs-all) and *one-versus-one* (also called all-vs-all and all-pairs) (Zimak 2006; Rifkin *et al.* 2003). One-versus-Rest (OVR) is the simplest scheme that trains an independent binary classifier for each class using all documents from that class as the positive examples and combining all the other classes into a single out-of-class, or negative category (Zimak, 2006). The one-versus-one (OVO) scheme trains a binary classifier to distinguish between each pair of classes (trained on a subset of data containing only those classes). This is a more powerful method than OVR as it captures the local interaction between every pair of classes. However, it can produce incoherent outputs such as class A over class B, class B over class C and class C over class A (Zimak, 2006).

While many authors have compared these schemes, the results are inconclusive with Rifkin *et al.* (2003) finding no significant difference in the performance between OVR and OVO. Often, a more practical concern is the computational complexity (Hoens *et al.*, 2012). For a dataset of M classes, OVR trains M models in comparison to M(M-1)/2 in the OVO scheme. The OVO scheme is therefore generally considered more expensive, and sometimes prohibitively so; but while each OVO model uses only a small subset of the data (containing only two classes), each OVR model uses the full dataset. Depending on how sensitive a particular algorithm is to the number of samples, the OVO might be faster (Hsu and Lin 2002; Rifkin *et al.* 2003). A final point to consider is that because OVR compares each class to all of the rest combined, it presents drastic class imbalance in the binary problems; even in balanced datasets (Hoens *et al.*, 2012).

Class imbalance is a major problem in classification (not just in the OVR decomposition) (Forman 2007; Hoens *et al.* 2012). The difficulty stems from the induction bias of most ML algorithms. Presented with a dataset with a severely underrepresented class, most algorithms will ignore the minority class as they can achieve high accuracy by always predicting the majority class (Hoens *et al.*, 2012). The effect of class imbalance becomes even more problematic in multiclass data according to Hoens *et al.* (2012) who further state that while class imbalance is relatively well studied (and understood) in binary problems, it has been largely overlooked for the multiclass problem.

The predominance of one of the classes in the training data leads to that class dominating the modelling process. The reason that this is more problematic for multiclass classification is because in the binary scenario the classes are complementary meaning that even if an algorithm is unable to learn the minority class adequately, its knowledge of the majority class may enable it to correctly classify minority class records by virtue of not being in the majority class (i.e. does not require any knowledge of the minority class). Moreover, feature selection methods tend to be dominated by a high number of strongly predictive features from easy classes, while ignoring the admittedly poorer features needed to discriminate the more difficult classes. In general, a class becomes easier to learn if more data is available meaning that the minority class is intrinsically harder, contains less discriminative features and may be overlooked by the feature selection process (Forman, 2007). Class imbalance can be addressed by adjusting the penalisation of minority class errors (cost-sensitive learning) or by artificially balancing the data through either under-sampling the majority class or oversampling the minority class (Kobayashi *et al.*, 2018).

A fundamental assumption in classification is that the past is a reasonable indication of the future. In practical terms this means that not only must the training data be representative of the problem setting, but it must be consistent with data that can be expected in the future (Alpaydin, 2010).

These requirements are formally expressed as the *independent, identically distributed* (IID) assumption made by many ML techniques. The condition of independence requires all training instances to be independent of each other. Namely, that they should represent unrelated, separate events such as the tossing of a coin (outcome of each toss is independent of all previous tosses). An identical distribution requires all training instances to be sampled from the same data generation process. In other words, that there are no overall trends or changes in the distribution (StatisticsHowTo, 2016).

However, this is rarely true in real world applications (Ganiz *et al.* 2010; Darrell *et al.* 2015). In reality, data often contains duplicates and repeated measures violating the independence assumption as well as changes in the underlying distributions (Witten *et al.*, 2011). These changes can be caused by many factors including sample selection bias and non-stationary environments (temporal or spatial changes). The most challenging is what Moreno-Torres *et al.* (2012) calls concept shift, when the fundamental relationship between the input and class variable changes i.e. changing the definition of a class label (e.g. the changing perception of what construes offensive language).

According to Moreno-Torres *et al.* (2012) there is a lot of terminological confusion surrounding the description of these distributional changes. In this thesis *data drift* is used to refer to gradual distributional changes over time while *data fracture* is used to describe more drastic, sudden changes leading to two distinct distributions before and after some fracture point. Cieslak and Chawla (2009) calls this the point of failure in the classifiers' predictions and would typically require training a new model. While many models may still work well if the changes are fairly subtle, drastic changes will require training a new model on updated data (Witten *et al.*, 2011).

Despite these violations, it remains a very useful assumption as it greatly simplifies the underlying mathematics (Ganiz *et al.* 2010; Darrell *et al.* 2015). Methods using the IID assumption can still provide very good results as long as the practitioner remains aware of the potential implications of these violations.

Machine learning is a very broad field with numerous different techniques, strategies and methods that have been successfully applied in a wide range of applications. Selecting between these can pose a challenging task, especially for novice practitioners (Wuest *et al.*, 2016). Empirical comparisons between the wide variety of methods have shown that each performs best in some, but not all, situations (Maimon and Rokach, 2010). This phenomenon is known as the *no-free-lunch the-*

*orem* which was formalised in Wolpert's famous 1996 paper where he demonstrates that no model can beat random guessing in all possible scenarios (Domingos 2012; Wolpert 1996).

This is based on the fact that ML is an inductive process and must make assumptions to be able to generalise beyond the training sample. The performance of a model depends on how well its inductive bias matches the properties of a particular dataset so that for any model, there exists a dataset for which it is accurate and another for which it is poor (Maimon and Rokach, 2010). Of course, one is not typically concerned with all possible scenarios but rather with one particular dataset for which there can theoretically be a "best" model. For this reason, Stapor (2018) states that it does not make sense to try and prove that one method is superior on average (nor should such claims be believed). Instead the focus should rather be on finding the conditions in a specific problem that makes one method perform better than another (such as class imbalance, dataset size, application domain).

In practice it is therefore recommended to evaluate several reasonable alternatives and select the one that performs best for that particular problem on that particular dataset with the selection of viable alternatives being guided by literature according to the prior knowledge of the specific problem and data characteristics (Raschka 2015; Alpaydin 2010). The no-free-lunch theorem does not only hold for the algorithm but for the entire modelling process, including all the data preparation and preprocessing decisions as these also make assumptions about data.

Something else to consider when comparing different models is complexity. Higher model complexity is associated with additional costs requiring more processing power, computational time and storage space making simplicity a desirable trait. This preference is often expressed through *Occam's Razor* which basically states that if two models perform equally well, the simplest should be preferred (Witten *et al.*, 2011).

The preference for simplicity is well-founded as complexity may cause models to overfit which hurts the generalisability of models beyond the training data (Witten *et al.*, 2011). However, care must be taken to not apply Occam's Razor blindly.

Domingos (1998) states that there are two possible interpretations (and applications found in literature) of Occam's razor of which only one is true. The first, that simplicity in itself is a modelling objective and should be pursued for reasons of interpretability and efficiency - is correct. But Domingos (1998) cautions against the second, the incorrect assumption that simplicity leads to better performance and demonstrates both empirically and theoretically that this is not true as complex problems warrant complex solutions (Domingos 2012; Domingos 1998).

It should be noted that while a large part of ML is automating data analyses, it is not intended to replace humans but rather to augment them (Maimon and Rokach 2010; Alpaydin 2010). Domain expertise remains incredibly important and ideally SMEs should be involved in every step of the process from data preparation, through modelling decisions and especially evaluation (Maimon and Rokach 2010; Alpaydin 2010; Witten *et al.* 2011).

## 3.2 Data Quality and Source

While ML is frequently applied to datasets that do not meet the normal standards of data quality (Edwards *et al.*, 2008), it does not mean it is insensitive to the effects of it. In fact, Mozetic *et al.* (2016) demonstrate that data quality, and specifically annotation quality, is the most significant determiner of model performance, much more influential than even the choice of algorithm or preprocessing methods.

In classification, the algorithm learns by example meaning that the quality of learning can only ever be as good as the quality of the training data (garbage in gives garbage out) (Maimon and Rokach, 2010). The problem is that the labels are dependent on the subjective judgement of humans who do not always agree with each other, or even with themselves (Mozetic *et al.* 2016; Lewis *et al.* 2004). The authors in Mozetic *et al.* (2016) suggest that the main reasons for disagreements are: the inherent difficulty of the task - especially when evaluating borderline cases; domain specific vocabularies where different words mean different things to different people; concept drift (e.g. what constitutes offensive language may change with time); or simply human error resulting from low quality work. They recommend monitoring the annotation quality throughout the labelling process so that problems can be detected early on and addressed (Mozetic *et al.*, 2016).

They evaluate data quality by assessing the extent of agreement between human annotators by duplicating 15% of the original data to be labelled twice by either different or the same annotator (unbeknownst to the annotators). They used two measures, namely *inter-annotator agreement* and *self-agreement* (Mozetic *et al.*, 2016).

The inter-annotator agreement evaluates the difference in labels assigned to the same document by different annotators. These inconsistencies provide an indication of the objective difficulty of the task rather than the data quality. Moreover, the authors found that this provides the upper limit of classification performance for that dataset (unless the self-agreement is too low) (Mozetic *et al.*, 2016).

The self-agreement evaluates the difference in labels assigned to the same document by the same annotator. This provides a good measure by which to identify low quality annotators. While poor self-agreement can lower the performance limit (by introducing unnecessary inconsistency in the data), good self-agreement cannot increase the performance beyond the limit imposed by the inter-annotator agreement. However, if the data is labelled by only one annotator, the performance is limited only by the self-agreement, but in this case the model is no longer learning the true "average" classifications but rather learning the classification process of that particular person (Mozetic *et al.*, 2016).

The authors in Mozetic *et al.* (2016) evaluated 17 datasets of various languages using multiple annotators on each. They report self-agreements varying from below 30% to above 80% and inter-annotator agreements varying from below 20% to above 60%. These results are not the exception. According to Lewis *et al.* (2004) the consistency of annotations has been shown to vary considerably in several studies on different datasets. This shows the unfortunate reality of the inconsistent and low-quality data that can be expected in practice. Apart from the labelling errors, the more general concerns of data quality such as incompleteness, noise, duplication and inconsistency are also relevant in classification (Maimon and Rokach, 2010).

The authors in Mozetic *et al.* (2016) recommend excluding poor quality data on the basis of low self-agreement as the model will be hurt more by their inclusion than it would gain from the additional training data. However, this provides a significant challenge in industry as often no surplus labels are available with which to evaluate the annotator agreement scores. As alternative Aljumaili (2016) recommends evaluating the data quality using meta data fields. Because metadata defines, describes and constrains data, any discrepancy, violation or anomalies are good indicators of poor data quality. For example, violations of data type or value constraints (e.g. text in numeric field, telephone number with too few digits), and data anomalies (e.g. order of magnitude differences in numbers, or single character descriptive text fields) can be indicative of poor-quality records. The downside is that such measures do not actually consider the data content or its labelling quality, addressing only the more general data quality issues heuristically.

### 3.2.1 Choice of Data

Machine learning research is largely driven by the available data (Lewis *et al.*, 2004). A common challenge in machine learning research is the acquisition of relevant and sufficient training data (Wuest *et al.*, 2016). There are three sources of data typically used in ML studies, namely synthetic data, publicly available benchmark datasets and industry data.

Synthetic data is generated artificially by researchers to try and mimic the real-world in a controllable manner with the specific purpose of testing some hypothesis (Demsar, 2006). The advantage of such data is that there is no limit to the amount of data available meaning an infinite amount of independent tests can be performed without the need for cross-validation or other subsampling techniques (Bellinger *et al.*, 2012). Moreover, it allows the researcher to precisely specify the conditions of the experiment and in so doing eliminate any unknown external effects that may affect the results unintentionally (Rodriguez *et al.*, 2010). While this can provide valuable insights to the effects of specific test conditions, it may result in unrealistic datasets not representative of the complex interactions found in noisy, real-world datasets. The results of such studies are as dependent on the researcher's ability to emulate real data as it is on the modelling decisions under consideration

(Demsar, 2006). For this reason, Alpaydin (2010) strongly recommends avoiding synthetic data as far as possible and rather to focus research on real-world datasets (benchmark or industry databases) collected under real-life circumstances.

Publicly available, benchmark datasets such as the UCI repository have many benefits as they enable researchers to use real-world data without each having to perform costly data collection and labelling. More importantly however, it enables replication, transparency and provides a baseline on which new algorithms can be evaluated (Lewis *et al.* 2004; Salzberg 1997). However, Lewis *et al.* (2004) cautions that just as a model can overfit to a specific sample of training data, so the broader research community can overfit to these benchmark datasets. This is confirmed by Salzberg (1997) who further states that these repositories are not representative of the larger population of classification problems but are in fact a very limited sample of problems. They both recognise the continual need for studies performed on new, real-world datasets to ensure a representative set of problems are addressed by academia as a whole (Lewis *et al.* 2004; Salzberg 1997).

Industry datasets provide the most realistic representation of not only real-world data but also provides insight to the real-world problems and domains (Lewis *et al.*, 2004). However, there are very real disadvantages to industry data as well. The biggest challenge concerns the limited availability. Companies are wary of letting their data into the public domain and while a researcher might be allowed to use their data and even publish the results, they will rarely be permitted to publish the accompanying datasets as well preventing replication or verification by other researchers (Kostoff, 2005). Moreover, this means that the labelling effort cannot be shared by the broader research community leading to many researchers performing unsupervised learning on unlabelled data or following inadequate annotation procedures.

Apart from the general quality considerations discussed above, document classification also faces many additional challenges due to the characteristics of text.

## 3.3 Text Characteristics

Natural languages are incredibly expressive and are capable of conveying even the most complicated, abstract ideas in a variety of ways (de Vos *et al.*, 2016). However, they are also highly unstructured, deeply ambiguous and tremendously complex data sources making automated processing a challenging task. The most challenging aspects of natural language processing is the ambiguity as well as the high dimensionality and sparsity.

### 3.3.1 Ambiguity

Even when processed by humans, language can be highly ambiguous and frustratingly vague. Ambiguity refers to anything that can have multiple meanings (de Vos *et al.*, 2016). There are many potential sources of ambiguity - poorly constructed sentences, context-dependent word definitions, synonymy, polysemy, figurative imagery and subtle variations in language (chronologically, geographically or by domain) to name a few.

To illustrate just how ambiguous language can be, one needs only consider the fact that a large portion of contractual litigation revolves around ambiguity in text. Despite contracts being specifically drawn up for clarity and precision, many disputes still arise from different interpretations of these contracts.

Humans are often able to disambiguate the correct sense through context or knowledge of the real world: either because only one interpretation is meaningful, or because one interpretation is dramatically more plausible than the others. However, computers have no real-world knowledge with which to judge the sensibleness of the various interpretations and are subject to much more ambiguity than might be immediately obvious to a human reader.

There have been attempts to encode background knowledge into computers to lessen this effect, most notably through ontologies. Ontologies try to formalise the domain of discourse. These can range from a synonym list to more complex conceptual models that describe the various entities, their properties and conceptual relationships between them (Chougule and Chakrabarty, 2009). These can be general-purpose or specific to a particular domain (Poli, 2003). While these have been successfully applied in some fields, Chougule and Chakrabarty (2009) note that both the creation and upkeep of ontologies is a very time-consuming and error-prone process presenting a barrier to many applications.

A further challenge in text analyses is the extensive use of non-literal, or figurative language. Figurative language is very much ingrained into all forms of written and spoken language, even scientific literature. For example, in ML literature a "black box model" is used to describe an uninterpretable model, where the imagery refers to the inner workings being hidden from view as if in a black box. However, a computer will be unable to distinguish between this and the literal use of either the colour black or a physical box.

Furthermore, language is not a stationary thing. It changes with time, region, culture and domain. These challenges become even more significant when analysing at a word level which is often the case as discussed in the next section.

### 3.3.2 Dimensionality and Sparsity

The core of machine learning is finding recurring patterns that can be used to create a model which applies to new data (Witten *et al.*, 2011). Although the various algorithms differ in precisely how and when in the modelling step they do this, all of them utilise some form of similarity measure between documents to find consistent similarities between documents from the same category, and consistent differences between documents from disparate categories. This information is then used to construct a model that classifies new documents according to their level of sameness (or differentness) to the various categories (Maimon and Rokach, 2010).

The biggest problem in such analyses are things which look the same but aren't and things which look different but aren't. This is an especially significant challenge in text due to the ambiguity (discussed above) and the extreme variability of natural language. Natural languages have an unbounded dimension as a potentially infinite variety of possible expressions exist. Due to the large number of ways to communicate the same idea it is highly unlikely that any text segment longer than a few words will have been used exactly before. Even the same author presenting the same idea within a short time-lapse, is highly unlikely to produce an exact duplicate of an earlier text without actively seeing it; even if that is their intention (de Vos *et al.*, 2016).

Computers have no faculty for similarity and are only capable of binary comparisons, namely perfect match or not. For example, given the inputs: (I) “The pump broke.” (II) “The pump broke” (III) “The pump is not working.” and (IV) “Die brandstof is op.” a computer will assess them all as equally unalike; even though I and II differ only with a period, while I and IV are not even the same language and refer to completely different subjects. However, despite this a human will easily recognise I and II as equivalent, I and III as related and IV as completely dissimilar. One of the obvious reasons for this is the reoccurring elements in related texts while there is little overlap between unrelated texts. Therefore, it is common practice to evaluate text according to the linguistic elements it contains as this allows computers to perform partial matching.

Despite the infinite dimensionality of language considered at the document level, all the expressional variety of text is achieved by different combinations of an essentially finite set of linguistic elements, such as characters, words, phrases etc. There are much fewer elements than there are ways to combine them. Therefore, many authors suggest only considering the statistical distribution of these distinct elements to reduce the dimensionality to a more manageable level. However, each reduction in dimensionality is accompanied by an inevitable loss in meaning. The most common way to do this is through the *bag-of-words model* which represents each document as an unordered set of words (Alpaydin 2010; Reese *et al.* 2017).

Evaluating documents at a word level offers a significant reduction in dimensionality, however it can still be prohibitively large. According to de Vos *et al.* (2016), there are approximately 100 000 words used in everyday English which still provides a significant dimensionality problem. Furthermore, text is also very sparse as each document will typically contain only a fraction of the corpus vocabulary. This leads to an incredibly sparse data distribution as the representation of each document (as a function of the corpus vocabulary) will be mostly empty. Both the dimensionality and the sparsity increase exponentially for higher level-representations such as n-grams (overlapping phrases containing n words).

Word appearance in language is characterised by Heap's Law and Zipf's Law shown on the right and left of Figure 3.1. Heap's Law shows the vocabulary growth (number of distinct words) corresponding to the corpus growth (total number of words) (Serrano *et al.*, 2009). Firstly, this shows that the most dramatic growth happens initially meaning that even relatively small datasets are likely to have high dimensionality. Secondly, the continuous growth means that any addition of new data will increase the corpus vocabulary meaning that a classification model must be able to handle words unseen during training.

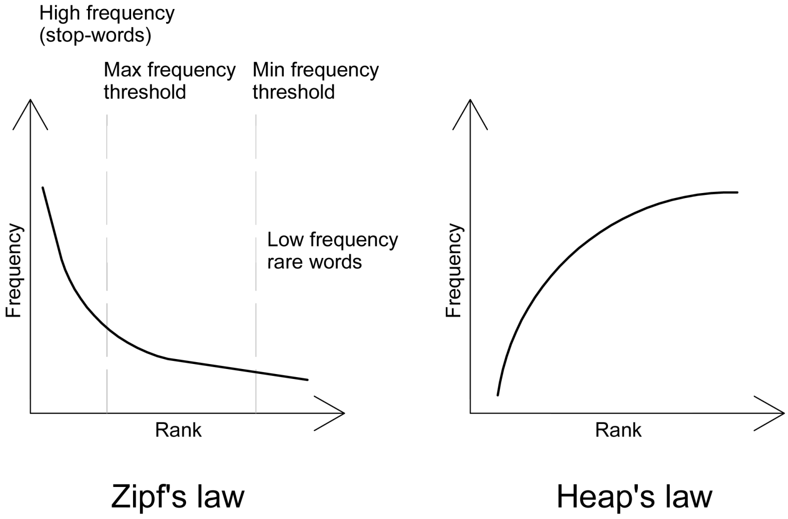


**Figure 3.1:** Text characteristics: Zipf's Law and Heap's Law

Zipf's Law states that the frequency of a term is inversely proportional to its frequency rank. This means that the most frequent term will appear approximately twice as often as the second most frequent term and three times more often than the next (Serrano *et al.*, 2009). This leads to a highly skewed distribution of words where only a few words occur frequently, while the majority of the words are very rare (Forman, 2007). For instance, in English the 10 most frequent words (the, be, to, of, and, a, in, that, have, I) make up 25% of all written text, while on the other end of the distribution 50% of all words occur only once - even in very large corpora[2]. These words, also called hapax legomena, make up only 5% of all written text but constitute a significant portion of the dimensionality (Witten *et al.* 2011; Press 2011).

Statistically speaking, the high frequency words are more significant. However, these are typically stop-words which add very little content information and serve more as grammatical construct. Their frequency is such that it can sometimes dominate frequency-based schemes and obscure much more significant, actual content-based words. On the other hand, the low frequency words, and especially the single occurrence hapax legomena, have much lower statistical significance as an algorithm cannot ascertain a pattern from a single point (Witten *et al.*, 2011).

Therefore, many researchers suggest removing both the high-frequency stop-words and the low-frequency rare words to reduce the dimensionality and improve the information density of the data in a process called frequency thresholding (Section 3.7).

A final characteristic of text is word-burstiness which refers to the notion that words are more likely to reappear in a document it has already appeared in, compared to its overall frequency in the corpus (Serrano *et al.*, 2009). This means that the first occurrence of a word in a document is the most informative.

## 3.4 Data Representation

At the core of machine learning is its ability to learn from data. For this reason, a critical step in the learning process is defining an appropriate representational framework in which to process the data (Witten *et al.*, 2011). This is not a trivial exercise as the choice of representation inevitably biases the learning scheme in a manner which could either enhance or limit performance (Witten *et al.*, 2011). The training data must be presented in a computer-suitable format. The standard

[2] Although considered a general phenomenon, the precise statistics will differ slightly from corpus to corpus. These statistics were computed from the Oxford English Corpus: a large collection of 21st century English from a wide variety of sources (literary novels, newspapers, blogs and more) and at the time contained more than 2 billion words (Press, 2011).

way to do this is by expressing the data in a matrix-like data structure called the vector space model (Sinoara *et al.*, 2017).

### 3.4.1 Vector Space Model

The Vector Space Model (VSM) is an algebraic model that represents textual information as a numeric vector to facilitate computer analysis. The dimension of the vector space corresponds to the number of features used to describe the text. In the usual bag-of-words implementation, this is the number of unique words found in the data-set, namely the corpus vocabulary. The training data is transformed to vector space by representing every document as a function of the words found in that vocabulary. This creates a matrix-like data structure where each row $i$ is a document and each column $j$ is a word from the corpus vocabulary. This representation is also called a Document Term Matrix (DTM) (Gentzkow *et al.* 2017; Reese *et al.* 2017).

So, for a training corpus consisting of $N$ documents containing a total of $p$ distinct words, the corpus vocabulary: $V = (w_1, w_2, ..., w_p)$ provides the matrix column indices. Each document $d$ can then be transformed to vector space: $V(d_i) = \overrightarrow{d_i} = [x_{i1}, x_{i2}, ..., x_{ip}]$ to populate the rows of a matrix to form the document term matrix:

$$DTM = \begin{bmatrix} \overrightarrow{d_1} \\ \overrightarrow{d_2} \\ ... \\ \overrightarrow{d_N} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & ... & x_{ip} \\ x_{21} & x_{22} & ... & x_{2p} \\ ... & ... & ... & ... \\ x_{N1} & x_{N2} & ... & x_{Np} \end{bmatrix} \tag{3.4.1}$$

Each matrix element $x_{ij}$ "measures" the presence of the $j^{th}$ word in the $i^{th}$ document. These feature values (matrix elements) can be encoded in various ways with the most common being Binary Occurrence, Term Frequency and Term Frequency Inverse Document Frequency (TFIDF) (Reese *et al.*, 2017).

**Binary occurrence features:** considers only the presence or absence of the $j^{th}$

word in the $i^{th}$ document using a binary representation such that:

$$x_{ij} = \begin{cases} 1 & \text{if document i contains word j} \\ 0 & \text{otherwise} \end{cases} \tag{3.4.2}$$

This representation ignores duplicates and has the effect of weighting each term in a document as equally important (Reese *et al.*, 2017).

**Term Frequency Features:** evaluates the number of occurrence of every word in a document under the assumption that the most frequent terms in a document are probably also the most relevant to that document. This can be thought of as the local frequency (document level). The simplest formulation of this evaluates the number of times that term $j$ appears in document $i$ namely:

$$x_{ij} = TF_{ij} = count(\text{word j in document i}) \tag{3.4.3}$$

However, although it is widely accepted that there is some correlation between the term importance and its frequency in a document, there is no consensus over the exact nature of this correlation with some authors suggesting sub-linear formulations instead. The most popular of these is through the use of Log Frequencies such that:

$$x_{ij} = log(1 + TF_{ij}) \tag{3.4.4}$$

where the logarithmic function lessons the effect of high frequency terms while still maintaining the positive correlation. In other words, while recognising that a word occurring 10 times more often than another is probably more relevant, it rejects the notion that it is 10 times more important. (The addition of 1 prevents zero-division for terms which are absent in the document.) (Marzec *et al.* 2014; Reese *et al.* 2017)

**Term Frequency - Inverse Document Frequency (TFIDF) Features:** evaluates the number of occurrences of a word in a document, normalised by the total number of documents that contain the word in the general corpus. It operates under the assumption that words which are common in every document will not provide much discriminative power. Then each element is:

$$x_{ij} = TFIDF_{ij} = TF_{ij} * IDF_{ij} \tag{3.4.5}$$

where TF is the term frequency (or one of its variants) which favours locally frequent words while IDF is the Inverse Document Frequency which scales down globally frequent words and is typically calculated from:

$$IDF_{ij} = log\left(\frac{N}{1 + n_j}\right) \tag{3.4.6}$$

where $N$ is the number of documents in the corpus and $n_j$ is the number of documents containing word $j$. In the same way as above, the logarithm is used to lessen the scale of the weighting and the addition of 1 in the denominator prevents zero-division. (Marzec *et al.* 2014; Reese *et al.* 2017)

Once again, there are numerous different schemes used in literature to calculate TFIDF. Some authors only scale the IDF component logarithmically (Sinoara *et al.*, 2017), others scale both TF and IDF logarithmically (Marzec *et al.*, 2014), others scale TF by document length (Rajpathak *et al.*, 2012) and many do not specify which formulation was used (Chen and Nayak, 2007).

All of the above formulations (and their variants) are regularly used in literature with no universally accepted, superior representation. For this reason, Forman (2007) recommends evaluating multiple schemes to find the best one (time and resources permitting), or otherwise, to choose a reasonable method based on literature.

In general, TFIDF is the most popular text representation scheme used in literature (Allahyari *et al.*, 2017). It is a fairly intuitive scheme and has been shown to achieve some impressive results. However, this is by no means a guarantee that this, or any other method preferred in literature, is the most appropriate for any given situation and Witten *et al.* (2011) cautions against neglecting more situation dependent, practical considerations.

Binary occurrence features provide the lowest level of information, TF lies somewhere in the middle and TFIDF features provide the highest level of information as it considers both the local (TF) and the global (IDF) characteristics of the data. Theoretically, the higher level of information provided by the TFIDF scheme should enable superior performance. However, if the additional information offered by higher level schemes are not relevant to the specific application, it could hurt rather than improve performance.

Finally, there are also some algorithms that require specific inputs. For example, Bernoulli Naïve Bayes require Boolean features. This can be circumvented by using some threshold value above which a TF or TFIDF feature value is mapped to one and below which it is mapped to zero. However, it is not clear whether this will provide any significant advantage over the Binary occurrence features (Witten *et al.*, 2011).

The schemes discussed up to now all present different variations of the bag-of-words model which characterises text by the words that appear in them. This is, by far, the most common scheme used in literature, but it is not the only one worth considering (Reese *et al.*, 2017). A common extension of this is the inclusion of higher order n-grams, which considers overlapping phrases of length n rather than individual words (which are unigrams). This preserves some level of word-order information while retaining the simplicity of the bag-of-words model. There is some contention in literature regarding the usefulness of higher-order n-grams which leads to an exponential increase in both the dimensionality and sparsity which can sometimes degrade the performance (Gentzkow *et al.*, 2017). However, while researchers may disagree on the principle of higher-order n-grams, according to Tan *et al.* (2002) it is widely accepted that n>3 is not useful and may even decrease performance. An important point made by Bekkerman and Allan (2004) is that higher order n-grams should only be considered as extension to the standard unigram (i.e. combined with and not as replacement) as the exclusion of unigrams generally hurts the performance. All of the above formulations (Binary, TF, TFIDF) are applicable to n-grams as well.

## 3.5 Bias and Variance Trade-off

The goal of supervised learning is generalisable predictive performance, namely to learn a classification function that will be able to predict the labels of new unseen data and not only memorise the provided training data (Maimon and Rokach, 2010). There are three sources of error that can affect the generalisation performance; the error due to bias, error due to variance and the irreducible error due to noise (Alpaydin, 2010).

In classification the noise refers to labelling inconsistencies whether due to undefined boundary cases or human error (Witten *et al.*, 2011). While some level of noise is inevitable, this does not mean that the irreducible error cannot be addressed. The inconsistencies due to the undefined boundary cases (inter-annotator agreement) should not be removed as this presents an inherent probabilistic component in the target objective due to the subjective nature of labelling (which is what is being modelled). However, if the labelling inconsistencies caused by error can be identified (e.g. by using self-agreement), these can be removed from the dataset to reduce the amount of noise in the data. It forms part of the irreducible error term because once included in the training set it cannot be reduced by improving the modelling process. As discussed in Section 3.2, these inconsistencies provide the upper limit of performance (Alpaydin, 2010).

It should be noted that this performance limit refers to the generalisable performance. It is possible to find a model that fits the data exactly, however because this would lead to also modelling the sample-specific error, such a model is likely to have poor generalisability. This is an example of an over-fitting model (Alpaydin, 2010). This is why it is typically recommended to evaluate the model performance on a separate sample of data (called a hold-out test set) unseen by the algorithm during training. This will provide a better indication of the generalisable performance as it is assumed that while the underlying predictive function will be consistent in different samples, the noise will vary from sample to sample.

The variance refers to the model's sensitivity to the training data. A high variance model is over-sensitive to the training data and will model the particular characteristics of that specific sample (including the noise). A model that suffers from high variance is typically over-fitting meaning that it has a poor generalisation performance as the model is dependent on the specific sample of data and will fluctuate between datasets. Alternatively, a low variance model will ignore the training data making the same prediction regardless of the input. This is also a high bias model. Bias refers to the data-independent assumptions made in the modelling process that cause systematic errors if incorrect. High bias can lead to models under-fitting the data due to its insensitivity to not only the noise but also the important features (Alpaydin, 2010). Such a model will also have poor generalisation performance.

The dilemma of trying to simultaneously lower the bias and the variance of a model is called the bias-variance trade-off because typically reducing one will lead to an increase in the other. Both are a function of complexity as typically the variance can be reduced by decreasing the complexity of models considered, while the bias can be reduced by increasing the complexity (Maimon and Rokach 2010; Alpaydin 2010). In order to learn a model that is applicable beyond the provided sample of data, a small amount of both bias and variance is required and the challenge is to ensure that the complexity is no more and no less than is required (Occam's Razor).

The bias and variance trade-off is not only applicable to model building but affects the model evaluation as well (Santafe *et al.*, 2015). Just as the modelling process can be over-sensitive to the training data, so the evaluation process can be over-sensitive to the testing data. When evaluating a model on a hold-out test set, one is not interested in the performance on that particular sample but using that to try and estimate the generalisation performance. However, it is possible that a specific sample could accidentally be much easier or harder to classify due to the particular documents contained in it meaning it is not a good indication of the generalisation performance. If an evaluation procedure is too dependent on the particular test-set being used, it suffers from high variance while if the generalisation error is always over or underestimated it suffers from high bias. This presents a significant challenge because while the variance of the modelling procedure can be reduced by increasing the amount of training data; and the variance of the evaluation procedure can be reduced by increasing the amount of testing data; typically there is only a finite amount of data available for both (Beleites *et al.* 2013; Santafe *et al.* 2015). Learning curves provide a way to diagnose the main source of error in a model and to evaluate the benefit of increasing the amount of training data (Raschka, 2015).

## 3.6 Learning Curves

Learning curves depict the performance of a model as a function of the amount of training data (Beleites *et al.*, 2013). The leftmost diagram of Figure 3.2 shows the general shape of such a curve. It starts off with a steep incline where a small addition of data translates to large performance gains. When there is too little data available, the algorithm has not received enough data to learn the concept fully and each additional instance provides new information and a substantial increase in the performance. However, this relationship does not increase indefinitely. After some threshold is reached, the algorithm has received a reasonably representative set of examples to learn the underlying concept and the curve starts to plateau with larger and larger amounts of data needed to achieve even small performance gains (Hastie *et al.*, 2009). This is known as the *law of diminishing returns* (Witten *et al.*, 2011).

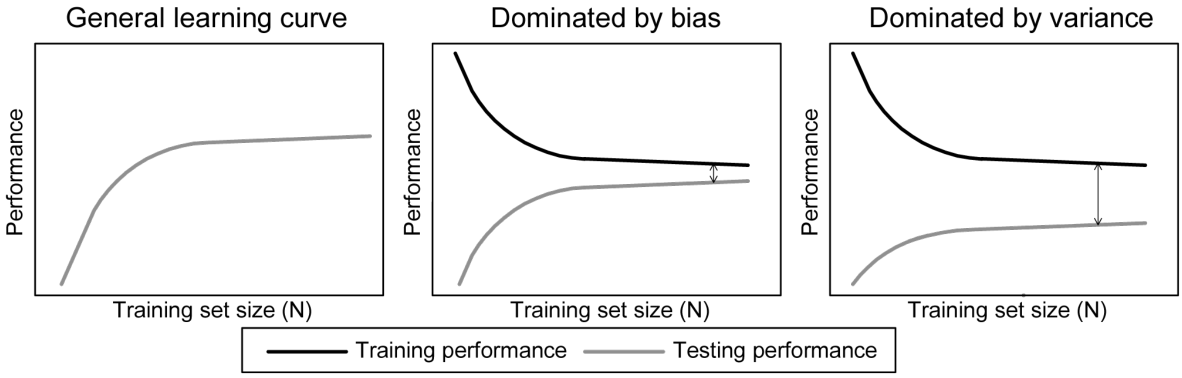


**Figure 3.2:** Typical learning curves

However, the true generalisation performance cannot be evaluated directly and only estimated from the empirical performance evaluated on a sample of data.

Therefore, it is useful to look at the learning curves evaluated on a sample of data. These are called conditional learning curves (Beleites *et al.*, 2013). Conditional learning curves are created by evaluating a model trained on an increasing portion of the training data to see how the performance is affected by the sample size (Witten *et al.*, 2011).

The middle diagram in Figure 3.2 shows the typical learning curves for experimental models with the upper black curve showing the training performance (evaluated on the same data used for training) and the lower grey curve showing the testing performance (evaluated on a separate hold-out sample of data unseen during training) (Luz 2017; Beleites *et al.* 2013; Ritchi 2019; Raschka 2015). From this it can clearly be seen that the hold-out test performance is a much better approximation to the true performance curve.

If the amount of training data is too small, the noise is indistinguishable from the signal and the algorithms tend to overfit to the specific characteristics of the sample leading to a high training performance but low generalisability beyond that sample (as indicated by the low test performance) (Beleites *et al.*, 2013). As the amount of training data increases, the model becomes less likely to overfit as the underlying function begins to dominate the incidental patterns in the data. While ignoring the noise leads to lower training performance, the generalisability of the model increases as indicated by the increasing test performance. While the training performance will always be higher than the testing performance, these will continue converging as the amount of data increases until the curves start to plateau (Luz 2017; Ritchi 2019).

When performing experiments, learning curves can provide valuable insight to the benefit of collecting more data as well as diagnosing the main source of error in models to guide further improvement efforts (Luz 2017; Ritchi 2019; Raschka 2015). If the model performance has not yet started to plateau at the maximum sample size, the collection of more data is highly recommended as the return on investment is high. If the amount of training data is adequate, a model should be sufficiently beyond that threshold to where the curves start to plateau (Hastie *et al.*, 2009). However, that does not mean more data will not be beneficial. Depending on whether the model suffers predominantly from variance or bias, it may or may not be worthwhile to collect more data.

The area between the training and testing curves (arrows on the figure) is indicative of the model variance (Raschka, 2015). If, like in the middle diagram, the curves converge, the model has low variance. For these models, the error is likely dominated by bias and it is unlikely to benefit much from the addition of more data. Such models are likely underfitting the data and may benefit from increasing the complexity to reduce the bias (e.g. by considering more complex algorithms, more complex feature sets or reducing the regularisation). (Luz 2017; Ritchi 2019; Witten *et al.* 2011; Raschka 2015)

On the other hand, models for which the curves do not converge (like in the rightmost diagram) suffer from high variance and are likely to benefit from the addition of more data (Raschka, 2015). A big difference between the training and testing performance is an indication of poor generalisability meaning that the model is likely overfitting the data. This can be addressed by decreasing the complexity (e.g. considering simpler algorithms, simpler feature sets or more regularisation) or by the addition of more data if the researcher feels that the complexity is warranted (Luz 2017; Raschka 2015). Of course, high bias and high variance are not mutually exclusive, and it is possible for a model to suffer from both. A low training performance is always indicative of high bias, regardless of the curve convergence.(Luz 2017; Ritchi 2019; Raschka 2015)

A final thing to note is that while increasing the amount of training data does generally increase the generalisation performance, it can have the opposite effect if the quality of the additional data is too low. This is illustrated by Mozetic *et al.* (2016) who found that the addition of poor-quality data (as indicated by low self-agreement) actually lead to a decrease in the test performance.

## 3.7 Basic Techniques

Machine learning is a very broad field with numerous different techniques. This section provides an overview of the basic techniques used in document classification. An important point made by Alpaydin (2010) is that practitioners should not code their own implementations of these techniques, but rather use one of the several publicly available libraries to do so as these have been optimised and validated by a large number of experts. This also enables replication of results (Alpaydin, 2010).

### 3.7.1 Data Cleaning and Preprocessing

Preprocessing concerns all the cleaning and preparation steps required to get data into a format suitable for modelling. Preprocessing consumes the bulk of the effort and time going into the entire data analysis; comprising up to 60% of the total data mining project according to Cios *et al.* (2007) and Kurgan and Musilek (2006). While it has no tangible business outputs, it is a critically important step with Witten *et al.* (2011) stating that the effort going into this process pays off many times over as industry data is typically of disappointingly low quality.

The success of the various preprocessing methods is very much data and application specific. While literature can provide valuable guidance, there is no substitute for good data understanding. It is an iterative process requiring substantial trial-and-error as there is typically no way to know before-hand how any given method (or combination of methods) will work for a specific implementation (Witten *et al.*, 2011).

The high-level preprocessing methods common to all machine learning applications (not just text) is discussed first followed by the more text-specific transforms.

#### 3.7.1.1 General Considerations

Any analysis should start with an exploration of the available data to determine the applicability of assumptions such as IID and to get a feel for the specific data characteristics such as class distribution or potential data quality issues. In this phase visual tools such as graphs can be very helpful. At this point it can be useful to look at all of the available data, not only the text fields. In the case of maintenance records, this includes the entire WO database because while the descriptive text fields are the focus of the analysis, the structured fields could provide valuable insight to the quality of the records and might even be useful features in themselves (Aljumaili, 2016).

The goal is to refine quality of the training data to maximise the performance that can be achieved. Ideally this would include evaluating the labelling quality according to the self-annotator agreement defined in Section 3.2. Other things to look at include the duplicates, missing data and outliers.

Duplicate records can result from data handling errors. This can be problematic as the repetition gives these records more significance than they actually have (Witten *et al.*, 2011). These are typically discarded (Maimon and Rokach, 2010). Most datasets have at least some missing values. It is important to consider the potential reasons for missing values when deciding how to handle them. For instance, missing values resulting from poor data collection discipline may be indicative of a low quality record, but if omitted due to irrelevance (e.g. failure symptom field for an inspection task) this might not be the case (Witten *et al.*, 2011). Outliers in any of the fields may also be indicative of poor quality records, however Witten *et al.* (2011) cautions that these should only be discarded if it is certain that they are not a valid anomaly of process but the result of some human mistake or data handling error. Domain expertise is invaluable in this phase as they can provide insights to the various reasons for any of the above issues (Witten *et al.*, 2011).

#### 3.7.1.2 Text Cleaning

The purpose of data cleaning is to remove any irrelevant information (extraneous variations in the text) as these may obscure meaningful patterns in the data and inflate the dimensionality unnecessarily leading to poor classification performance (Kobayashi *et al.*, 2018).

Text is a very high dimensional, noisy data source. While using the BOW model to analyse the data at a more granular level reduces the input variation (Section 3.3), it is still a very high dimensional source with a lot of noise even at a word-level. Additional clean-up is frequently required to lessen this variation and increase the signal to noise ratio, although inadvertently some signal will be lost as well. The most typical steps deal with special character encoding/decoding, capitalisation, punctuation, numbers and spelling-errors as well as more advanced steps such as stop-word removal, stemming, feature selection, feature scaling and dimensionality reduction

While text cleaning tools can be created from scratch using Regular Expressions[3], due to the growing popularity of text-mining applications, many machine learning platforms already have dedicated text preprocessing tools for this purpose which can be used as is or customised to suit specific needs.

### Special Characters Encoding

Character encoding refers to how computers store text, namely the mapping between the actual characters that occur in text and the numeric representation scheme used by the computer (McCallum, 2012). Essentially, the encoding is a table that translates bytes into human readable characters. As the computational field progressed, multiple different text encoding schemes have been developed such as ASCII, Code Page 1252, Mac Roman, Shift-JIS, Unicode and many more. These schemes differ in the exact mappings used as well as the character-sets they address.

If the encoding scheme used to store the data is unknown, the data could be misinterpreted or even corrupted if decoded with the wrong scheme. All the encoding schemes transform text to a computer readable bit sequence. However, if using the wrong scheme, a bit-sequence could be mapped to an entirely different character. Or, if no such bit-sequence exists in that scheme, a program might silently discard the invalid sequence or replace it with some place-holder replacement symbol such as □ or � (Zentgraf, 2015). This can be a problem because often, a researcher will receive plain text files and frequently do not know the correct encoding scheme used. While there are tools available that can help detect the encoding scheme used, McCallum (2012) cautions that these are not perfect and recommends inspecting snippets of the text manually to evaluate the viability of the characters produced.

[3] A sequence of characters that defines a search pattern that is more powerful than exact match searches in text due to its use of several wildcard notations (for example “\d” can be used to search any single digit, “\D” any single non-digit, “\w+” one-or-more word characters and “\d2, 5” between two and five digits).

For English applications, once the correct encoding has been identified, it may be a good idea to convert any non-ASCII characters to the nearest equivalent ASCII characters (e.g. naïve to naive) in a process called Transliteration by Marzec *et al.* (2014). In English, diaereses are mostly found in foreign names and loan words such as façade and naïve. But even here they do not change the meaning and are, in fact, neglected by some authors which would lead to a computer regarding inconsistent usage of "façade" and "facade" as two distinct words. Most programming languages have tools for this e.g. Python Unidecode package which translates any Unicode to the closest ASCII representation. For more detailed discussion on dealing with encoding problems refer to McCallum (2012) and Zentgraf (2015).

### Capitalisation

Capitalisation can add a significant portion of the dimensionality of text without adding much meaning. On a character level it already doubles the number of distinct features (from 26 to 52) which becomes even more extreme on a word level. For example, sentence starting capitalisation carries no content information but will drastically increase the dimensionality if e.g. "The" is treated distinct from "the". For this reason, case information is often removed by transforming all text to lower case in a process called case-normalisation. However, there are rare cases where capitalisation can be meaningful. Some words, called capitonyms, change meaning when capitalised and can, for instance, help to distinguish between the Polish nationality and polish used for shining shoes. (This is just for English, in some languages capitalisation might be even more significant e.g. German which capitalises nouns (Witten *et al.*, 2011).)

### Punctuation

Punctuation symbols are also often removed to reduce the dimensionality of the data. Otherwise mid-sentence occurrences of the word "pump" will be evaluated as distinct from sentence ending "pump." (and all other variations due to exclamation marks, commas, brackets etc.).

Common practice is to remove punctuation during the tokenisation process. However, how this is done is not always so trivial (Blamey *et al.*, 2012). At the most basic level, the punctuation can simply be removed without replacement (I);

replaced with a single white-space (II); or replaced with a unified punctuation placeholder variable (III). All of these will get different things right and wrong. For example, in the contraction "can't" the first method (I) would give <cant> which is probably preferable to <can>, <t> provided by the second (II) as this would make it impossible to distinguish from its opposite "can". However, for the decimal point in the number "9.21", the first method (I) yields 921 which changes the order of magnitude completely.

For this reason, some authors suggest applying different rules for different punctuation marks for example McKenzie *et al.* (2010) who retains all apostrophes before the letters s, t and d.

### Numerals

In many applications, numeric terms are regarded as adding irrelevant variation and therefore dimensionality. Kobayashi *et al.* (2018) recommend removing numbers to improve model efficiency unless there is reason to believe doing so will hurt the performance.

Similar to punctuation, this can be done through simple removal, replacing with white-space or by replacing with a numeric placeholder (i.e. only retain that it was a number, not what number). For example, if classifying books into topics, a high prevalence of numeric terms might be indicative of a text book rather than a romance novel. However, treating each unique number-term (including page numbers, publication dates and every example) as a distinct feature is less likely to be useful and will quickly make the dimensionality intractable. As with most modelling decisions, the choice of how to (and whether to) remove numbers is application specific (Kobayashi *et al.*, 2018).

### Language Mistakes: Spelling Errors and Typos

Over and above the incredible complexity and variety of perfect language, language is seldomly perfect. Formal text sources such as academic journals or newspaper articles undergo rigorous editing and contain much less error, although even then it prevails[4]. Informal texts, which are usually not edited, have a much higher prevalence of spelling, grammar and miswording errors.

For this reason, some authors suggest using spell-checkers to correct misspelt words during the preprocessing phase. While it may, on average improve results for generic language usage, Forman (2007) warns that automatic spelling correction mistakes may outweigh the benefits as unfamiliar terms are forced to the "nearest" known word which may be incorrect[5]. This is an especially bad idea for technical documents which may have many uncommon, domain-specific words not found in a typical vocabulary (Forman, 2007).

### Part-of-Speech Tagging

In Part-of-speech (POS) tagging, words are tagged with their respective parts of speech such as noun, verb or adjective (Maimon and Rokach, 2010). There are multiple off-the-shelf POS taggers available for English. These are usually trained on manually tagged corpora such as the Brown corpus meaning that essentially they are supervised classifiers. This tries to capture the syntactic relationships between words to provide a more expressive representation than the traditional BOW to, for example, distinguish between "pump" used as a verb, noun or adjective. However, this is not a trivial process and incorrect tags may hurt rather than help the performance with Allahyari *et al.* (2017) stating that POS tagging is a tedious and error-prone process. A final point that should be noted is that POS taggers use the full sentence information to identify the correct part of speech. This means that documents containing sentence fragments might not be compatible with standard taggers (Mukherjee and Chakraborty 2007; McKenzie *et al.* 2010).

### Tokenisation

Tokenisation refers to splitting up of text into distinct tokens to be used as features

[4] Such as the humorous case of "The Wicked Bible" which accidentally printed: "Thou shalt commit adultery." as the 7th commandment by accidental omission of the word "not" in one of the early King James versions (Eisiminger, 1989).

[5] Such as the poor student, in whose entire thesis document an incorrect spelling of "tests" was changed to "testes".

in the vector space (Allahyari *et al.*, 2017). The most typical version of this is splitting text on whitespace to get a word-level representation, namely the bag-of-words model. However, tokenisation does not have to refer to words, it can split on any character (or sequence of characters, or punctuations etc.) as long as the end result is an unordered set of tokens. Moreover, the tokens do not have to be distinct. For instance, the n-grams representation overlaps to both sides as each word interacts with the word before and after it (Witten *et al.*, 2011).

The tokens are represented in vector space by some numeric value which is typically somehow related to the frequency of occurrence. There are numerous options but the most common are binary occurrence (1 if present and 0 if absent), integer frequency count (number of occurrences) or more complex scaled versions such as log frequency or TFIDF (Section 3.4).

The above sections already mentioned some of the issues regarding tokenisation - although simple in concept, the implementation is often not as trivial. Choosing exactly how to split on which characters can be difficult (Witten *et al.*, 2011). Moreover, while one of the most ubiquitous techniques in text-mining literature, the majority of researchers do not report the actual process used leading to high uncertainty when trying to compare studies or emulate the results (Blamey *et al.*, 2012).

**Stemming**

Stemming tries to reduce the extraneous dimensionality caused by the inflection of words by reducing words to their base form, or at least trying to (Kobayashi *et al.*, 2018). Inflection refers to changing the form of a word to express differences in tense (work, worked, working), number (pump, pumps), gender (waiter/waitress) and more.

Inflection can add a lot of expressive nuance to words which aids in expressivity and particularity, but often at the cost of generalisability. Frequently, the inflectional difference between words are a language construct that does not actually change the meaning of the word in a major way. Computers, only capable of exact match will evaluate the words “run”, “runs” and “running” to be completely distinct and as dissimilar as “run” and “stop”. By evaluating these as distinct words, not only do the dimensionality (and sparsity) increase unnecessarily, but performance can be adversely affected by the feature dispersion.

Stemming is most typically done through a set of heuristic rules such as removing trailing “s” from words. These methods are purely statistical and do not take the individual word, meaning or context into account at all. However, this can lead to situations of both over-stemming, merging distinct words to a common form (e.g. “news”/“new”), and under-stemming, not merging words together which should be (e.g. “mouse”/“mice”) (Jivani, 2011).

### Stop-word Removal

Stop-words are not only a meaningless addition to the dimensionality of text, they are sometimes even considered harmful as their high frequency can obscure actual significant patterns. These are typically removed using a stop-word list compiled for a specific language (Witten *et al.*, 2011). However, it is not always so straight-forward. For instance, removing the stop-words changes the meaning of the following sentences: “she was arrested” changes to “she arrested” and “the pump is not working” changes to “pump working”.

There are even some applications where stop-words prove the most important features, precisely because of their content independence and ubiquity. In authorship attribution for example, the content-heavy words vary highly according to different topics addressed by an author, but the function words stay relatively constant, posing as more of a stylistic feature (Witten *et al.*, 2011). However, this is a very specific scenario that is not frequently relevant to more general text classification tasks.

### Feature Selection

Feature selection can be both supervised or unsupervised depending on whether they make use of the class labels. Some of the most common unsupervised methods include stop-word removal (described above) and frequency thresholding which is used to remove both the most frequent and most rare terms above and below some user specified maximum and minimum threshold respectively. While any of the feature representations can be used, this is typically done using the document frequency (DF) under the assumption that a word appearing in all documents does not have discriminative power while words appearing in too few documents do not have the statistical validity with which to make inference (Kobayashi *et al.*, 2018). Maximum frequency thresholding will affect many of the same words as stop-word removal, but enables a more corpus specific elimination.

Supervised feature selection makes use of the class membership information to determine the predictive power of each feature. Here the focus is more on selecting the best, most informative features rather than on discarding the worst, uninformative features. Common methods here include Mutual Information, Chi-Squared, Gini Index and Information Gain (Kobayashi *et al.*, 2018).

### Data Scaling

Feature scaling comprise the set of techniques that can be used to scale features to a comparable range. This can be done per document (per row in DTM) or per feature (per column in DTM).

Document length normalisation prevents the features from longer documents dominating those of shorter documents by normalising each document vector according to its vector length. This can be done using either the Euclidean $L_2$ norm, or the Manhattan $L_1$ norm:

$$||\vec{x}||_{L_1} = (|x_1| + |x_2| + ... + |x_p|) \tag{3.7.1}$$

$$||\vec{x}||_{L_2} = \sqrt{(x_1^2 + x_2^2 + ... + x_p^2)} \tag{3.7.2}$$

The IDF transform can also be considered a feature scaling method. It accounts for the frequency component resulting from corpus frequency rather than document relevance by scaling each feature (column in DTM) by its inverse document frequency.

Finally, feature standardisation rescales the feature columns to that of a normal distribution centred around a mean of 0 and a standard deviation of 1. This retains outlier information but make models less sensitive to them. This is useful for distance based classifiers.

**Dimensionality Reduction**

While all the data cleaning steps discussed up to now can be considered dimensionality reduction methods, the word is more typically used to describe those techniques that transform the features to a lower dimensional space. One of the most popular methods for text is Latent Semantic Analysis (LSA) which performs a singular value decomposition of the document term matrix to reduce the dimensionality of the vector space. This reduces the dimensionality of the feature space and may even lead to improved performance through better feature representations. However, inevitably such a representation loses some descriptive detail and may not be interpretable any more (Edwards *et al.*, 2008).

### 3.7.2 Algorithms

An algorithm is the mathematical process which, when applied to training data, produces a model such as a classifier. It is useful to distinguish between the hyporameters, which the algorithm learns from the data, and the hyperparameters, which can be set (and optimised) by the practitioner (Tsamardinos *et al.*, 2015).

There are many different classes of algorithms that have been successfully applied for document classification including K-Nearest Neighbours, Naïve Bayes, Support Vector Machines, Decision Trees and Neural Networks to name a few (Lee and Yang 2009; Kobayashi *et al.* 2018). As per the no-free-lunch theorem (Section 3.1), there is no algorithm that is best for all situations as all of these make various assumptions about the data and the learning objective making them more or less applicable in various situations (Maimon and Rokach, 2010).

Naïve Bayes and Support Vector Machines are particularly popular due to their efficiency at handling the high dimensionality and sparsity that is characteristic of text (Kobayashi *et al.*, 2018)

#### 3.7.2.1 Naïve Bayes

Naïve Bayes is very popular due to its speed and simplicity. It is therefore often selected as a baseline method with which to compare more complex algorithms. Despite its simplicity, it can perform surprisingly well in certain applications showing comparable performance to more complex methods such as Support Vector Machines (Schneider, 2005).

It is a probabilistic method based on the application of Bayes Theorem and the so-called "naïve" conditional independence assumption whereby it assumes that the presence of features (words) in a document are independent of each other given the class (Mccallum and Nigam, 1998). While this assumption is typically violated in real-world applications, it enables incredibly efficient handling of high-dimensional data as the parameters for each feature can be learned independently (Mccallum and Nigam, 1998). Moreover, despite this violation it can perform surprisingly well (Witten *et al.*, 2011).

The classification function works by selecting the class that is most likely to have generated that document (using Bayes Theorem of posterior probability) and is also called a generative model. Variants of Naïve Bayes differ only in the assumption they make about the distribution of the features. The two most common are the Bernoulli Naïve Bayes and the Multinomial Naïve Bayes which assume a multivariate Bernoulli and multinomial distribution respectively (Schneider, 2005).

#### 3.7.2.2 Support Vector Machines

Support Vector Machines (SVM) are particularly well suited to the characteristically high-dimensional nature of text data as it is able to utilise the natural sparsity of text to avoid a dimensionality crisis (Allahyari *et al.*, 2017). Despite the no-free-lunch theorem, several studies have demonstrated SVMs consistently outperforming other models with some authors proclaiming it as the state-of-the art, industry standard (Mertsalov and McCreary, 2009).

SVM is a distance based, discriminatory classifier that looks for a decision surface that separates the classes in an n-dimensional hyperplane and maximises the margin of separation between them (Baharudin *et al.*, 2010). It is very efficient because only the documents closest to the decision surface are used to create the model; these are called the support vectors.

SVM is a linear classifier, but it can perform non-linear classification by using the kernel trick. While higher-order kernels have been used to great effect in other domains, they have not been shown to provide any discernible performance benefits for text classification leading only to an undesirable increase in model complexity (Lewis *et al.* 2004; Leopold and Kindermann 2002). Due to its efficiency at handling high-dimensionality, it is often claimed that SVM does not benefit from feature selection, but Forman (2004) states that this not true.

### 3.7.3 Hyperparameter (Model) Optimisation

The hyperparameter optimisation can have a drastic impact on the model performance. However, it is important to note that the hyperparameters are not limited to algorithm parameters but include any variable in the modelling process that is not estimated directly from the data (Tsamardinos *et al.*, 2015). This includes all user decisions such as the selection of preprocessing transforms and parameters.

Due to the complex interactions between the various parameters, algorithms and data there is no way to know beforehand which combination will yield the best results (Marzec *et al.*, 2014). For this reason, the optimisation typically comprises a trial-and-error process whereby different parameter combinations are implemented and evaluated according to the target objective (in this case classification performance) to select the best combinations (Witten *et al.*, 2011).

There are different optimisation strategies used in literature including *one-factor-at-a-time*, *exhaustive grid-search* and a *randomised-search procedure*. In all of these methods the researcher must first define the parameter search space to explore by selecting a set of parameters and a reasonable range of values for each.

As the name suggests, one-factor-at-a-time keeps all but one parameter steady to optimise each one-by-one. The problem with this method is that the combination of such separate optimisation outcomes is not guaranteed to yield optimal results (Baharudin *et al.*, 2010) as this method ignores the complex interactions that are known to exist between parameters (Alpaydin, 2010).

To combat this effect, the grid-search varies all parameters together instead of one at a time to perform an exhaustive search of every parameter combination in the search space. According to Bergstra and Bengio (2012), this is the most widely used strategy in literature but suffers from the curse of dimensionality as the search space grows exponentially with the number of parameters considered. For this reason, grid-searches quickly become computationally infeasible meaning that only a small subset of parameter-combinations must be selected by the practitioner. Therefore, some practitioners suggest performing a grid-search on a smaller subset of the data to ease the computational load. However, Forman (2007) cautions that the optimality of parameters may be dependent on the amount of data so that this can lead to the selection of suboptimal parameters.

Alternatively, Bergstra and Bengio (2012) recommends performing a randomised-search procedure which evaluates only a random sample of points on the grid under the assumption that there is a close-to-optimal region in the search space that will yield comparable results to the single optimal point.

The advantage of this method over the more common exhaustive grid-search is two-fold. Firstly, a much larger range of parameters settings can be explored for a fraction of the computational effort. This is especially important for inexperienced practitioners as selecting the most important hyperparameters and a specific range of values to consider can be a daunting task. It is not always apparent which parameters will have the largest impact on performance nor which specific set of parameter-values are likely to contain the optimum (Bergstra and Bengio, 2012). There are resources available that attempt to guide practitioners in selecting parameter-grids by specifying the most influential parameter(s) for each algorithm and providing a reasonable range of values for each. However, Bergstra and Bengio (2012) found that parameter influence (and optimal value-range) is very much data-dependent, a finding that is supported by the conflicting recommendations sometimes found in literature. Furthermore, despite the fact that complex interactions are known to exist between the various parameters, these resources frequently address only one aspect of parameter optimisation at a time (e.g. only the estimator parameters or only a single aspect of the preprocessing decisions). The combination of these separate optimisation outcomes is not guaranteed to yield optimal results (Baharudin *et al.*, 2010).

Secondly, because it randomly samples parameter combinations from the provided parameter space, the range of values to evaluate for each parameter can be provided as a continuous distribution. This enables a much finer optimisation than is possible for grid-search which can only evaluate a discrete set of values. This enables the randomized-search procedure to explore a much wider range of the important parameters while being relatively unaffected by the inclusion of unimportant parameters. This is because unlike grid-search, multiple parameters can be changed for each iteration. The grid-search wastes a large portion of effort keeping the important parameter values constant while testing different values of the unimportant parameters. In contrast, the randomized-search procedure varies both at once meaning that more values can be tested for the important parameter and is largely unaffected by simultaneously varying the unimportant parameter values. (Bergstra and Bengio, 2012)

While it is reasonable to doubt the effectiveness of this procedure as it does not evaluate each grid point, Bergstra and Bengio (2012) shows that in many instances, the randomised-search procedure is able to find models that are as good or better than the grid-search applied over the same search space. Overall, trying only 60 points seems to be sufficient. Zheng (2015) confirms that the randomised-search procedure is as good as an exhaustive grid-search in many instances and further explain the selection of 60 trials. While the full discussion is beyond the scope of this study, Zheng (2015) summarises their explanation by stating that *"if at least 5% of the points on the grid yield a close-to-optimal solution, then random search with 60 trials will find that region with high probability (95%)".*

### 3.7.4 Evaluation

As mentioned in Section 3.5, when evaluating supervised models the concern is with generalisation, namely estimating the performance that can be expected upon deployment for all data, not just the available training data (Maimon and Rokach, 2010). In doing so there are various things to consider including the evaluation metrics and evaluation approach.

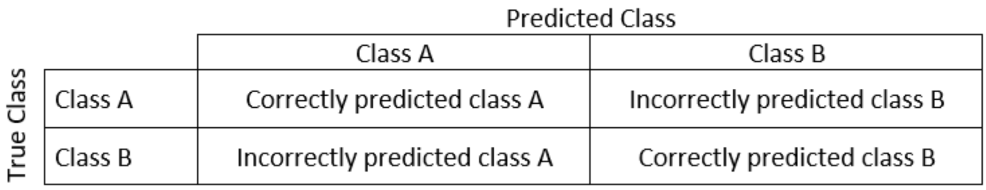
### 3.7.4.1 Evaluation Metrics

The most straightforward metric is Accuracy (or its complement misclassification error) defined as the proportion of documents labelled correctly (eq 3.7.3). (To distinguish the specific metric from accuracy used in the general sense to mean correctness, the metric is referred to with a capital letter.)

$$Accuracy = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = 1 - MisclassificationError \tag{3.7.3}$$

Accuracy is an intuitive and easy to understand metric, but it can be dangerously misleading in situations of class imbalance. For example, in a binary classification problem with a 90/10 class imbalance, a trivial classifier which always predicts the majority class will achieve a 90% Accuracy despite being a useless model. For this reason, it is widely recommended against to report only the Accuracy of models. Regardless, it remains one of the most widespread metrics with many studies still reporting Accuracy in isolation. Even if data is balanced it is still good practice to look at more than one metric to get a better understanding of the model performance as each gives a different perspective of the errors made.

Numerous other evaluation metrics exist to quantify the model performance. Classification results are often summarised using a confusion matrix which provides the number of correct and incorrect predictions for each class and from which the other metrics can be computed. An example can be seen in Figure 3.3 where each row represents the true class membership and each column the predicted class membership to not only indicate the amount of errors but also give insight into the type of errors made.



| | | Predicted Class | |
|---|---|---|---|
| | | Class A | Class B |
| True Class | Class A | Correctly predicted class A | Incorrectly predicted class B |
| | Class B | Incorrectly predicted class A | Correctly predicted class B |

**Figure 3.3:** Confusion matrix schematic

From this it is useful to distinguish between True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) for each class. For example, for class A:

- TP: number of documents correctly predicted as class A
- FP: number of documents incorrectly predicted as class A
- TN: number of documents correctly predicted as not class A
- FN: number of documents incorrectly predicted as not class A

It is desirable to maximise the diagonal elements of the matrix as these represent the correct predictions (TP). The off-diagonal rows provide the FN and the off-diagonal columns the FP. (Of course, the TP of one class forms part of the TN for all the other classes).

The confusion matrix can be used to calculate the Precision and Recall for each class. Precision determines what proportion of documents classified into class A are truly class A and can be calculated from the confusion matrix by taking the diagonal element of each class as a percentage of its column total as shown in eq 3.7.4. Recall determines the proportion of documents from class A that are correctly classified and can be calculated from the confusion matrix by taking the diagonal element of each class as a percentage of its row total as shown in eq 3.7.5. While both metrics maximise the TP, Precision penalises FP and Recall penalises FN.

$$Precision = \frac{TP}{TP + FP} \tag{3.7.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.7.5}$$

Ideally you want to maximise both of these measures for all classes, but often maximising one will lead to minimising the other. The cost of different types of error are not always equal, sometimes Precision (which penalises FP) may be more important than Recall (which penalises FN) or vice versa. For instance, in email-spam detection it is far more costly to delete a non-spam email than to let a few spam emails into the inbox.

In cost-sensitive learning where one type of error is more costly than another, this trade-off between Precision and Recall is evaluated by calculating the harmonic mean between them using a metric called the F-beta score:

$$F_{Beta} = (1+\beta^2)\frac{Precision * Recall}{\beta^2 Precision + Recall} \tag{3.7.6}$$

where $\beta$ refers to the relative importance of Recall over Precision. In the event that Precision and Recall are weighted equally, namely $\beta = 1$, it reduces to the F1-score as can be seen in eq 3.7.7. This is also called the balanced F-score and can be further simplified using the above definitions of Precision and Recall as shown below:

$$F_1 = 2\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{3.7.7}$$

The F-score is widely recommended for text-classification, especially in the case of class-imbalance where it provides a better view of the total performance than Accuracy. There also exists graphical methods to evaluate models, such as Area Under the Curve (AUC), but the extension to multiclass is not trivial (Santafe *et al.*, 2015) with no accepted method to do so in literature (Sokolova and Lapalme, 2009) and is therefore not presented here.

All the above measures determine the per-class performance, but can also be used to evaluate the overall model performance by taking either the micro- or the macro-averaged metrics. Micro-averaging determines the per-document metrics. It weights each document equally meaning that the majority class (which has more documents than minority classes) performance will dominate the score. Macro-averaging, on the other hand, determines the per-class averages. Therefore, it weights each class equally meaning that the influence of the minority classes on the score is upweighted beyond their proportionality in the dataset. For example, the difference between the micro- and macro-average is shown for Precision in equations 3.7.8 and 3.7.9.

$$P_{micro} = \frac{\sum \text{True positives}_{\text{all classes}}}{\sum \text{True positives}_{\text{all classes}} + \sum \text{False positives}_{\text{all classes}}} \tag{3.7.8}$$

$$P_{macro} = \frac{\sum \text{Precision for all classes}}{\text{Number of classes}} = \frac{\sum_{i=1}^{M} \frac{TP_i}{TP_i + FP_i}}{M} \text{ for M classes} \tag{3.7.9}$$

In balanced datasets these are equivalent, but in in cases of class-imbalance the micro-averaged scores can hide poor performance of minority classes (which are typically harder to classify), while the macro-averaged metrics are disproportionally influenced by the minority classes which make up only a small component of the dataset.

To some extent, the micro-average can be considered a better indication of the desired performance since the score is dominated by the majority of the documents. By definition, the majority class is the most frequently occurring class. Therefore, the performance of the majority class can be considered more important than that of the minority classes since you want the model to perform well for the majority of instances. However, it can hide unacceptably poor results for the minority class. The majority class performance may be more important overall, but there might be some threshold below which the minority class performance is unacceptable (e.g. a trivial classifier that always predicts the majority class can have a high micro-averaged score). For this reason it is generally recommended to consider multiple different metrics to get a broader perspective of the model performance. Overall, however, the choice of evaluation metric (and averaging method) should be application specific.

#### 3.7.4.2 Other Considerations

While the predictive performance (as quantified by the above metrics) is important, other factors to consider when evaluating models is the interpretability, efficiency and actionability (Witten *et al.*, 2011).

Interpretability (also called comprehensibility) refers to how well humans can grasp the model and understand the relationships it identifies. People are more accepting of what they understand making interpretability an important factor for industry adoption. Interpretable models can be reviewed by experts. Not only is this important for error-finding and validation, but it also enables the incorporation of expert feedback into the modelling process leading to improved models. Moreover, interpretable models open up the possibility of experts being able to identify and use the interesting patterns found by these models (as opposed to using just the models) (Maimon and Rokach, 2010).

Efficiency refers to the computational efficiency and concerns the memory, speed and complexity of the analysis. For applications requiring real-time predictions, such as fraud detection, efficiency is of utmost importance, but otherwise only limited by practicality. Actionability refers to the potential usefulness of the model to justify the data mining effort (Witten *et al.*, 2011).

#### 3.7.4.3 Evaluation Approach

The main approaches used for performance evaluation are training performance, hold-out test sets and cross validation. The training performance, also called the resubstitution method, is perhaps the most simple evaluation approach that uses all of the available data to first train and then test the model (Santafe *et al.* 2015; Varma and Simon 2006). Because the same data is used to train and test the model, this approach tends to be optimistically biased (underestimates the generalisation error due to overfitting) (Maimon and Rokach, 2010). With this approach, a trivial model that simply memorises the training data would get a perfect resubstitution score despite being useless beyond that specific sample (Santafe *et al.* 2015; Witten *et al.* 2011). For this reason, it is generally not considered to be an adequate approach. While the training performance can be useful to know (Section 3.6), it is generally recommended to evaluate the model on unseen data to emulate the deployment scenario such as the hold-out method (Santafe *et al.* 2015; Witten *et al.* 2011).

### Hold-out Method

The hold-out method, as the name suggests, holds out a subsample of the available training data to better estimate the generalisation error, namely the performance on unseen data. There are multiple ways in which this can be done including two-way, three-way and k-fold cross-validation (Santafe *et al.* 2015; Raschka 2015).

In a two-way evaluation, the data is split into two mutually exclusive sets: a training set used to create a model and the testing set used to evaluate the performance (Santafe *et al.*, 2015). However, most ML applications also involve an optimisation process in which hyperparameters are selected according to the performance as evaluated on the test set. While the two-way evaluation offers a drastic improvement to the resubstitution approach, it can also be optimistically biased as the test set is used repeatedly to tune the hyperparameters. Just as models can overfit to the training data, so the modelling process can also overfit to the testing dataset. Because the test set is used repeatedly in the hyperparameter optimisation process, it can no longer be considered a good imitation of generalisation performance on unseen data as the model may have been over-optimised to the characteristics of that specific test-set.

For this reason, it is preferred to split the data three-ways to create a training, validation and test set so that the model can be optimised repeatedly according to the performance on the validation set with only the final, optimised model evaluated on the hitherto unseen test set.

From the learning curves (Section 3.6) it can be seen that model performance is a function of the amount of training data, but Beleites *et al.* (2013) point out that building a good model is not enough; the performance must also be validated (Beleites *et al.*, 2013). The more data available for training, the lower the modelling bias meaning a better model can be trained. However, assuming a finite amount of data, an increase in the training sample equates to a decrease in the testing sample in the hold-out scenario. This increases the evaluation variance as the performance estimate becomes more sensitive to characteristics of a particular test set leading to a less reliable evaluation. Using a larger testing sample will reduce the variance and thus improve the validity of the evaluation; but will again result in a smaller training sample leading to a weaker model.

Cross-validation attempts to solve the above mentioned problems by using multiple training/testing subsets of the data to train multiple models and in so doing also get multiple performance evaluations. The most common formulation of this is k-fold cross-validation in which the data is divided into k equal folds (subsets). Then k models are trained using each fold as a test set and the remainder as training data. Each document is therefore used as a test document exactly once and used k-1 times as training (Santafe *et al.*, 2015).

This method allows models to use all of the data for training and all of the data for testing in an attempt to minimise both the bias of the model and the variance of the evaluation. Rather than having a single point estimate of performance it gives multiple values indicating the spread of performance that can be expected. It should be noted however, that the cross-validation is only used as performance estimate. The final model should be trained using all of the data. Because all of the performance estimates are of models trained on less data than the final model, assuming the general trend of increasing performance with more data, this means that the cross-validation estimate should be slightly conservatively biased (better than overconfident).

The choice of k is once again a bias variance trade-off. For larger k's the training sets become large meaning that the performance estimate bias is reduced and will be closer to the true value. However, this also means that the test set sizes will decrease meaning that evaluation variance will increase making the CV estimate less trustworthy. Also, the larger k is, the more models are trained leading to an increase in the computational and manual effort. On the other hand, smaller values of k mean that the training sample for each model decreases which result in poorer models and an increase in the conservative bias of evaluation (bigger difference between models evaluated and true model). But correspondingly, the test set increases which decreases the test variance improving the validity of the evaluation.

For all of the hold-out methods, the dataset can be separated using either a random or a stratified splitting strategy. In the random approach, the data is typically shuffled and split randomly into the number of sets required by the hold-out scheme. However, for imbalanced data, this can result in minority classes being excluded from some of the folds. In these situations a stratified scheme is recommended which maintains the class proportions in all of the folds (Raschka, 2015).

## 3.8 Validation

Apart from evaluating the specific scores of a classifier, it is important to consider the validity of both the modelling and the evaluation process as this pertains to the validity of the results.

### 3.8.1 Baselines

When evaluating models it is good to have a baseline with which to compare the results. Some authors suggest using the performance on publicly available benchmark datasets as a baseline to compare different methods.

However, good performance on a benchmark does not guarantee the superiority of a method, which might simply be particularly suited to the specific properties of that data. Lewis *et al.* (2004) cautions that industry as a whole can become overfit to benchmark data if used repeatedly to justify the preference of one method over another. Baharudin *et al.* (2010) further state that experiments performed by different authors cannot be compared due to the various incidental, “background conditions” extraneous to the particular algorithm or method under consideration that affect the results. Therefore it is good to have a problem-specific baseline that can be created by every author for their specific dataset. One such method is what is called “dummy estimators” in the Scikit learn documentation.

Dummy estimators are used to quantify how much of the model performance can be attributed to chance. A trivial model that simply assigns all documents to the majority class (which has frequently been used as an example in this thesis) is one such dummy estimator, in particular a Majority Class Dummy Estimator. It provides a useful baseline comparison in cases of class imbalance. Other possibilities include a Random Dummy Estimator and a Stratified Dummy Estimator. The Random Dummy Estimator assumes uniform priors for each class while the Stratified Dummy Estimator uses the observed class distribution to make data predictions. While none of these are useful models, they provide a useful baseline comparison which can be used to evaluate actual models.

### 3.8.2 Statistical Validation

Due to the experimental nature of ML, some researchers propose using various statistical tests to determine the significance of results (e.g. whether the performance gains observed for a new algorithm is statistically significant or due to chance).

However, there is some controversy surrounding the validity of such tests (Bergmeir and Benitez, 2012) as typical ML experiments violate many of the assumptions these tests make (such as the lack of independence between the training folds of a cross-validation) (Dietterich 1998; Stapor 2018). In fact, due to the frequent misuse and misinterpretation of statistical tests in literature, Demsar (2006) states that many researchers are of the opinion that significance tests should not be used at all.

Dietterich (1998) reports that the most popular method observed in ML literature, the paired t-test, is not suitable (due to high probability of making a Type I error) further stating that it should never be used. While this view is shared by several authors, there is much less agreement about what should be used in its stead and from the conflicting views in the literature surveyed, no definitive answer could be found.

However, according to Salzberg (1997), research focussed on feasibility do not require a statistical evaluation to be convincing. Moreover, Witten *et al.* (2011) states that for most practical applications simply choosing the method with the best cross-validated performance is sufficient; even if this difference is due to chance and not statistically significant. As mentioned previously, the objective is not to find the single "best" model, but rather to find one that is good enough and demonstrate this empirically. More important than proving the statistical significance of results, is ensuring the validity of the modelling and evaluation procedure used to obtain them to make sure the results are not overestimated.

### 3.8.3 Methodological Concerns

Several concerns have been identified in literature that pertains to the validity of the various steps typically found in ML projects. First is the concern of data leakage (also called data contamination) which occurs when the model has unfair insight to the test-set when training leading to overconfident performance estimates. This can be due to a wide variety of issues. The most drastic version of this is if the label is accidentally included in the training data leading to a trivial classifier "predicting" the label based on the label (Sapkota *et al.*, 2015). However it can also occur in more subtle ways.

According to Krstajic *et al.* (2014) a common mistake made in literature is to optimise the preprocessing transforms outside of the modelling process, namely before the hold-out data is separated, leading to an optimistic bias. Both Krstajic *et al.* (2014) and Tsamardinos *et al.* (2015) recommend that the selection of preprocessing transforms should be treated as hyperparameters and optimised with the algorithm hyperparameters in the cross-validation. This also enables the preprocessing transforms to be optimised for each algorithm. According to Baharudin *et al.* (2010) proper optimisation of the preprocessing transforms can have a big impact on the algorithm performance further stating that the predominance of SVM in literature is due to people comparing unoptimised versions of the algorithms. While this does not impact the validity of the results it can lead to selecting suboptimal models.

Moreover, in the typical scenario where cross-validation is used for both model optimisation (tuning the hyperparameters) and model evaluation, data leakage can also occur by manner of the algorithm over-optimising for the specific test set. This is no different to the repeated optimisation in a two-way split meaning that the performance estimate is no longer truly a hold-out evaluation leading to a dangerously optimistic bias. Therefore Krstajic *et al.* (2014) and Tsamardinos *et al.* (2015) recommend performing a nested cross-validation, namely a cross-validation within a cross-validation where the inner loop is used for optimisation and the outer loop is used for the performance evaluation. They recognise that this will result in a pessimistic bias due to the smaller amount of data used for training, but they state that this is preferable to an optimistic bias.

Another concern identified by Bergmeir and Benitez (2012) is the violation of the IID assumption and its impact on model evaluation. By splitting the data randomly (typically after shuffling) or through stratification you create an artificially IID train/test distribution where the model is evaluated on data consistent with the data it was trained on. While this would be the ideal situation, more frequent than not there is at least some form of data drift. They therefore recommend doing a blocked cross-validation (also called a grouped CV) whereby the data is split into more homogeneous blocks (based on some grouping) so that the classifier can be evaluated in a more realistic scenario where the training and test sets differ. While this can be done for any dependencies, they evaluate the impact of making chronological time-blocks and using these as the folds in a cross-validation. They find that while this method underestimates the performance, this was preferable to both randomised and stratified cross-validation which over-estimated the true performance (Bergmeir and Benitez, 2012). Lewis *et al.* (2004) does not use a cross-validation but proposes a similar concept for a three-way split saying that the chronologically last data should be kept apart for testing to emulate the deployment scenario where a model is trained on historical data and tested on future data.

Another concern noted by Forman and Scholz (2010) is the different ways in which the cross-validated F-score can be computed. They note that according to the definition of the F-score in eq 3.7.7 when averaging the results of a cross-validation this can be computed in one of two ways. The cross-validated F-score can either be computed as a single metric using the aggregated fold predictions ($F_{AGG}$), or by calculating the F-score for each fold and then averaging the results ($F_{AVG}$). From a literature survey they found evidence of both methods being used, but that many did not report which was used stating that the vast majority of researchers seemed unaware of this discrepancy. However, they note that there is a significant difference between these two methods and that $F_{AGG}$ is the least biased formulation and should always be used; especially in the case of class imbalance. This is because of the non-linear penalisation of error for very small classes leading to over-penalising mistakes on the minority class. This effect is lessoned by aggregating the TP over all the folds to remove the score away from the highly non-linear region near TP=0.

# Chapter 4

# Literature Review: Related Studies

This chapter reviews the text mining literature that is most relevant to the project at hand, both within and beyond the maintenance domain. Only industry focussed research, or at least those that consider real world data are included. It addresses the fourth and fifth project objectives to gain insight into the current state of research as it pertains to solving the specific research question as well as how it fits in to the broader machine learning and text mining literature. Along with the validity concerns identified in the previous chapter, the outcome of this is used to guide the experimental analysis.

It starts by considering the studies specific to the maintenance domain. Although initially limited to those concerning failure modes, too few studies could be found and so the scope was broadened to include all text mining literature that consider historical maintenance records. The next section considers the broader text mining literature, focussing specifically on data with similar properties as the maintenance records, particularly those with short document lengths. An important outcome of this chapter is to compare the common practices observed in the industry focussed literature with that recommended by the more academic literature.

## 4.1 Domain Specific Literature

Several authors confirm both the prevalence of and challenges associated with text-based maintenance records in a variety of industries including the automotive domain (Rajpathak and De, 2016), military helicopters (McKenzie *et al.*, 2010), power generation (gas and steam turbines) (Mukherjee and Chakraborty, 2007), the railway sector (Wang *et al.*, 2017), manufacturing (Sipos *et al.*, 2014), coal mills (Uz-Zaman *et al.*, 2015), water infrastructure (Edwards *et al.* 2008; Chen and Nayak 2007); public transport (Marzec *et al.*, 2014) and according to Mukherjee and Chakraborty (2007), Reeve (2016) and Devaney *et al.* (2005), almost all asset intensive and service organisations. Such data cannot be processed using traditional data analytics and require time-consuming and labour-intensive manual processing which few can afford.

With the exception of Reeve (2016), all the above mentioned authors identify some form of text mining as potential solution, although they acknowledge various domain specific challenges over and above the already significant difficulty of more standard text mining applications (such as high-dimensionality and sparsity) (McKenzie *et al.* 2010; Devaney *et al.* 2005). This includes short document length, class imbalance, non-standard English usage and very little domain-specific research which sets it apart from more general literature conducted on corpora of much better quality than is typical for maintenance records (Edwards *et al.* 2008; Rajpathak 2013).

### 4.1.1 Domain Specific Challenges

Due to limited input space (character limitations) and severe time-pressure under which data is collected, maintenance records are frequently much shorter than that typically considered in literature (Mukherjee and Chakraborty 2007; Chen and Nayak 2007; Devaney *et al.* 2005). Most TM research is focussed on documents with more than 100 words, such as the common Reuters-21578 dataset which has an average document length of 160 words (Timonen, 2012). In contrast Chen and Nayak (2007) report records ranging from 1-50 words and even more extreme Mukherjee and Chakraborty (2007) reporting texts of 5-10 words.

For many of the studies reported here, class-imbalance was a major challenge. Because the training data is dominated by examples from one class, models tend to favour this majority class and perform poorly on the smaller classes. For failure mode classification in particular, Wang *et al.* (2017) note that the performance on minority classes is also important for assuring the safety and efficiency of assets, not just the majority and middle fault classes. While the class imbalance will vary from dataset to dataset and according to the classification objective, all of Wang *et al.* (2017), Sipos *et al.* (2014), Edwards *et al.* (2008) and Uz-Zaman *et al.* (2015) identify varying levels of imbalance in their datasets.

While there is a substantial amount of literature and other resources available for natural language processing, these have been developed for Standard English (or standard language) and is not suitable for maintenance records due to the poor grammar, high proliferation of abbreviations and the unique and specialised vocabulary and syntax used by maintenance personnel that more closely resembles short-hand notation than standard English (McKenzie *et al.* 2010; Edwards *et al.* 2008; Rajpathak 2013).

Because the data is typically not collected by the end-user, it is often treated as a secondary task and recorded in fairly haphazard ways (Devaney *et al.*, 2005) with little to no quality controls in place during data collection (Edwards *et al.*, 2008). Variation in the input is extremely large, exhibiting all the expressive freedoms of natural language but none of the typical constraints of spelling or grammar (Devaney *et al.*, 2005). This leads to a high prevalence of language errors and incomplete segment fragments in the input data (Devaney *et al.* 2005; McKenzie *et al.* 2010).

The data further has a high proliferation of extremely terse, non-standard abbreviations and acronyms used inconsistently throughout the dataset (Mukherjee and Chakraborty 2007; Devaney *et al.* 2005; Edwards *et al.* 2008). These are not consistently marked with punctuation and may be specific to a machine, department or even an individual (McKenzie *et al.*, 2010). Rajpathak (2013) compares the number of unique abbreviations found in their maintenance dataset with that of a standard benchmark corpus finding that the maintenance records contain 107 unique abbreviations in comparison to 6 in a similar sized sample of the benchmark data. Added to this is the particularity of the maintenance domain. According to McKenzie *et al.* (2010), the majority of words used in maintenance records are domain specific. This relates to uncommon usage of general terms (such as seal referring to a rubber seal not an animal) as well as technical terms, jargon and abbreviations not found in general corpora. This prevents the usage of general-purpose dictionary or thesaurus-based resources (Devaney *et al.*, 2005).

However, unlike more generic text-mining tasks (such as sentiment analyses) there are also no subject specific resources like ontologies or benchmark datasets available since little research has been done for the maintenance domain specifically (Chen and Nayak 2007; McKenzie *et al.* 2010; Edwards *et al.* 2008). The lack of benchmark datasets limit researchers to industry data availability (Chen and Nayak 2007; McKenzie *et al.* 2010). As mentioned in Section 3.2.1, while there are many advantages to real-world data sets, there is also much less available, especially in the maintenance domain.

The implication of this is slower progress as few researchers have access to data and even fewer to labelled data as the labelling effort cannot be shared across multiple studies (Devaney *et al.* 2005; Edwards *et al.* 2008). Moreover, the studies with labelled data often have only small samples available and can typically only publish the results and not the datasets preventing replication and the establishment of strong baselines. According to Edwards *et al.* (2008), such studies become an investigation into what results can be expected from TM methods when the usual standards of data quality and size are not met.

### 4.1.2 Overview of Studies Considered

The potential value of the unstructured, free-text portions of maintenance records is well-documented in literature and has been used to create reliability models (Rajpathak and De 2016; Uz-Zaman *et al.* 2015), construct fault trees (Mukherjee and Chakraborty, 2007), identify best-practice repairs (Rajpathak, 2013), accurately predict maintenance budgets (Edwards *et al.*, 2008), reduce downtime and prevent failures (Devaney *et al.*, 2005), improve inventory and spare part management (Rajpathak *et al.*, 2012) and any number of other activities that enhance both the strategic and operational decision-making processes (Rajpathak and Chougule, 2011).

As mentioned before, there is limited research available on the topic of maintenance records, and even less dealing with failure mode classification. The most relevant, and which are considered the primary studies of this review, are those by Wang *et al.* (2017), Chen and Nayak (2007), Marzec *et al.* (2014), Edwards *et al.* (2008), Uz-Zaman *et al.* (2015), McKenzie *et al.* (2010), Mukherjee and Chakraborty (2007) and Devaney *et al.* (2005). Of these, only Wang *et al.* (2017) and Chen and Nayak (2007) consider failure modes while all of Marzec *et al.* (2014), Edwards *et al.* (2008) and Uz-Zaman *et al.* (2015) are concerned with distinguishing between corrective and preventive events (namely failure and non-failure). McKenzie *et al.* (2010), Mukherjee and Chakraborty (2007) and Devaney *et al.* (2005) do not involve any specific class distinction as they are concerned with information extraction rather than classification. However, they still provide valuable insights to the data cleaning and preprocessing of maintenance records that best deal with the domain specific challenges discussed above.

Where relevant, aspects of the studies by Bastos *et al.* (2014), Sipos *et al.* (2014) and the series of studies by Dnyanesh Rajpathak and other authors (Rajpathak and Chougule (2011), Rajpathak *et al.* (2012), Rajpathak (2013) and Rajpathak and De (2016)) are also briefly mentioned. However, due to notable differences between these and the above studies; they are only considered in a supplementary fashion and not discussed in great detail.

While the studies of Bastos *et al.* (2014) and Sipos *et al.* (2014) are also concerned with distinguishing between corrective and preventive events; their input data is different to the verbatim text records considered in this thesis and the primary studies. Bastos *et al.* (2014) consider only structured data using a combination of numeric and coded CM data along with the structured event data fields such as component ID. Sipos *et al.* (2014) do consider textual event data, but their input consists of automatically generated equipment logs. While the desired information is still implicitly buried in text, because they are machine-generated from a template, they are not natural language and contain much less variation than human-generated text. However, both give valuable insights to the modelling processes found within the AM domain.

Dnyanesh Rajpathak and several other authors performed a series of text-mining studies on a large sample of automotive text-based records: Rajpathak and Chougule

(2011), Rajpathak *et al.* (2012), Rajpathak (2013) and Rajpathak and De (2016). These are not considered as part of the primary sources due to their extensive use of high-level domain specific ontologies not typically available to industry practitioners (nor to this study in particular). However, they are a good example of the possibilities of text-mining and also the only studies that have been deployed in industry.

The lack of industry data is evident from the fact that two of the primary studies, Mukherjee and Chakraborty (2007) and Devaney *et al.* (2005), had no data available providing only a theoretical proposal with no experimental component to verify their claims. Apart from Marzec *et al.* (2014) and Uz-Zaman *et al.* (2015), all the primary studies considered datasets with fewer than 1 000 records. Even Marzec *et al.* (2014), with the biggest dataset at 15 000 records is not a massive sample in terms of text mining problems. Both Uz-Zaman *et al.* (2015) and Wang *et al.* (2017) evaluate different training-testing ratios to show how model performance improves with larger training sets (learning curves in Section 3.6) and all of Chen and Nayak (2007), McKenzie *et al.* (2010) and Edwards *et al.* (2008) confirm that their performance can be expected to improve with more data.

Of the primary studies that had data available, distinction can be made between those that had labelled and unlabelled data. Only Marzec *et al.* (2014) and Wang *et al.* (2017) had labelled data (labelled by subject matter experts) while the rest labelled data themselves or resorted to unsupervised methods.

In Edwards *et al.* (2008) the authors tried two approaches; first labelling the data themselves using a best-guess approach, and later using an unsupervised clustering algorithm to find the natural categories of the data. Due to the absence of SME labelled data, McKenzie *et al.* (2010) also labelled the data themselves but used it to train a supervised POS tagger rather than for classification. Chen and Nayak (2007) also resorted to unsupervised clustering methods. Uz-Zaman *et al.* (2015) labelled their data using meta-data fields (such as urgency or down-time information) to create a labelling filter that assigns labels heuristically. A similar approach was taken by one of the secondary studies, Sipos *et al.* (2014), who also created a meta-data filter but recognised that this was not an ideal approach due the imperfect approximations used to construct such a filter.

Of those who had labelled data available with which to perform classification, the majority looked at binary classification and specifically at distinguishing between corrective and preventive maintenance events (Marzec *et al.* 2014; Edwards *et al.* 2008; Uz-Zaman *et al.* 2015; Sipos *et al.* 2014; Bastos *et al.* 2014). Edwards *et al.* (2008) performed both multiclass and binary classification: a binary model trained on the records they labelled themselves and a multiclass classifier trying to predict the natural categories of the same records which had been "labelled" by an unsupervised clustering algorithm. They confirm that multiclass classification is a more difficult problem as the binary output variable simplifies the problem search space (Edwards *et al.*, 2008). Only Wang *et al.* (2017) considered multiclass failure mode classification. (Chen and Nayak (2007) also considered failure modes but tried to find them through unsupervised clustering.)

Both Edwards *et al.* (2008) and Chen and Nayak (2007) performed unsupervised clustering on their respective datasets to obtain their target objective: corrective vs preventive and failure modes respectively. While the clustering provided some interesting insights into their data and into the effectiveness of preprocessing techniques, both reported dissatisfaction at the natural categories provided by the clustering algorithm not matching the desired groupings. This is a frequent problem of unsupervised learning as the model cannot be guided to provide the output required by the researcher.

None of these studies addressed the label quality directly by means of the annotator agreement measures as recommended by Mozetic *et al.* (2016) (Section 3.2) since this would require additional labelling effort and most struggled to get even a single labelled dataset. However, they do not dispute the importance of data quality, and of the label quality in particular (Marzec *et al.* 2014; Bastos *et al.* 2014). And while Sipos *et al.* (2014) emphasise that labelling error could drastically affect the quality of the results, they recognise the practical constraints of data availability and that it may be infeasible to improve data quality. This is confirmed by Mukherjee and Chakraborty (2007) and Uz-Zaman *et al.* (2015) who state that models must be constructed from the available data to bridge the gap between the data that is available and the information that is required. Both Chen and Nayak (2007) and Edwards *et al.* (2008) state that SME labelled data would be preferable to their respective approaches but recognise the importance of using what is available. This is especially relevant in academic contexts (such as this) where researchers have limited interaction with subject matter experts.

#### 4.1.2.1 Performance Results

The performance of text-mining applications depends on a number of factors including the complexity of the target objective, the algorithms used, and the nature, quality and amount of data available (Marzec *et al.*, 2014). This makes it difficult to compare the results of different studies; not one of which used the same dataset. In fact, as mentioned before, comparisons are only reliable when based on experiments performed by the same author under carefully controlled conditions, which is not the case here (Baharudin *et al.*, 2010). Despite not comparing the actual scores achieved by these studies; it is evident that as could be expected, the best results are achieved by those who had access to large datasets and domain expertise.

In particular Marzec *et al.* (2014), who looked at binary classification of urban bus maintenance records into preventive and corrective events, achieved very good results. They had the largest sample of labelled records available (15 000 records) and the most extensive SME involvement; from labelling to data cleaning, preprocessing and model evaluation. It is also interesting to note that their dataset was in Polish making them the only non-English study. This may have had an impact on their success as Polish and English have significantly different grammars.

The objective of their study was to investigate the viability of the TM approach to see if existing methods are sufficiently accurate to use in business decisions. With an accuracy of 99% they concluded it was, but it should be mentioned that there is a possibility that these results are over-estimated due data leakage in their modelling process (discussed below) (Marzec *et al.*, 2014).

The importance of domain expertise is confirmed by all of Mukherjee and Chakraborty (2007), McKenzie *et al.* (2010), Marzec *et al.* (2014), Edwards *et al.* (2008) and Chen and Nayak (2007). While neither Edwards *et al.* (2008) nor Chen and Nayak (2007) were even able to obtain SME-labelled data due to the academic nature of their work, they both emphasise that expert involvement should ideally go beyond the labelling phase and comprise the entire process.

The studies by Rajpathak can also be considered highly successful as they were the only ones found that has been successfully deployed in industry. Like Marzec *et al.* (2014); they also had a large amount of data available (more than any of the primary studies) and extensive SME involvement; but most significantly they had access to high level domain specific ontologies they developed in previous research (Rajpathak and Chougule, 2011), namely a reliability ontology (Rajpathak and De, 2016), a diagnosis ontology (Rajpathak, 2013) and an Integrated Vehicle Health Management ontology (Rajpathak *et al.*, 2012).

While the value of domain specific ontologies is undisputed, and clearly evident from these studies, they are not typically available to researchers or industry. This is confirmed by McKenzie *et al.* (2010) and Chen and Nayak (2007) who further recognise the substantial effort and expertise required to create these manually limiting the practicability of this approach. The high prevalence of jargon and non-standard vocabulary found in these records limits the usefulness of general-purpose, linguistic ontological resources (Devaney *et al.*, 2005). Furthermore, even within the AM domain many of the terms are highly specific to a particular machine, set-up and industry and may vary significantly between companies, departments and even individuals requiring each practitioner to develop their own. In fact, despite using very similar datasets and considering only one asset type (automobiles), in the studies by Dynash Rajpathak each used a different ontology created specifically for that problem (Rajpathak *et al.* 2012; Rajpathak 2013; Rajpathak and De 2016). While this was facilitated by the ontology development framework they created in earlier work (Rajpathak and Chougule, 2011), they also recognise the substantial effort required to create and maintain these resources.

In the primary studies, Mukherjee and Chakraborty (2007) and Devaney *et al.* (2005) also considered ontology creation. Mukherjee and Chakraborty (2007) proposes combining WordNet (linguistic ontology) with a specific machine's Bill of Materials to create a machine-specific ontological resource but they do not verify this experimentally. To reduce the effort required by each practitioner developing application specific ontologies, Devaney *et al.* (2005) proposes using industrial standards to create more general-purpose ontologies that can be shared across a number of application domains and types of machinery. They work on the basis that all equipment share high-level systems and subsystems (such as hydraulics pneumatics and electronics) with machine independent characteristics (e.g. all hydraulic systems have hoses, pumps and fluid). However, they have also not verified this experimentally.

The emphasis on domain expertise reiterates the point that advanced analytical tools and methods are not meant to replace SMEs but rather to enhance their productivity and that obtaining their support and involvement can be very beneficial. One way to encourage stakeholder support, is to create interpretable models. People are more accepting of what they understand and while this is true for all domains, Bastos *et al.* (2014) and Sipos *et al.* (2014) emphasise that model interpretability is exceptionally important for the maintenance domain specifically, stating that it is crucial to be able to show how or why the model works. As mentioned in Chapter 2, a major challenge remains convincing managers to trust data over their own intuition (Baglee *et al.*, 2015). Using interpretable modelling methods mean SMEs can review and incorporate expert feedback into the modelling process leading to better and more valid models. Even when SMEs are not involved, according to Sipos *et al.* (2014), just knowing that the model is “interpretable”, and understanding which the most important features are, make experts more comfortable adopting it. This has an effect on the techniques used as dimensionality reduction methods like SVD or black-box algorithms such as Neural Networks reduce the interpretability of the models making them less acceptable to industry professionals.

None of the primary studies were particularly concerned with model efficiency other than the fact that the techniques must be able to handle the large number of features typically found in text data. While all text-mining applications require a degree of efficiency to handle the high-dimensionality of text data; this concerns the training rather than the prediction efficiency as none of the primary studies required real-time predictions.

Several authors recognised the value of sample selection. In Bastos *et al.* (2014), the authors delete records with missing values to improve the quality of the input data. In Sipos *et al.* (2014) and McKenzie *et al.* (2010) the authors discard certain records to increase the homogeneity of the input data. Sipos *et al.* (2014) select only records from a specific component in a specific machine while McKenzie *et al.* (2010) select only the records concerning inspections (discarding all failure events). McKenzie *et al.* (2010) compare the results of this more homogeneous group with that of their full sample and found that the performance improved significantly showing the value of such sample selection. In Edwards *et al.* (2008) and Chen and Nayak (2007) the authors discard all meta fields (coded inputs containing mostly identifiers such as LocationID and time-stamp data) retaining only the descriptive text fields. Edwards *et al.* (2008) further concatenate all the text fields into a single input to extend the document length to mitigate the short document length challenges.

### 4.1.3 Methodology

While the actual scores of the various studies cannot really be compared due the differences in their specific datasets and modelling objectives; it is useful to compare the various tools and techniques to discern what may be useful for the empirical analysis in Chapters 6-8[1].

Data cleaning and preprocessing is discussed in varying levels of detail in the different studies. Uz-Zaman *et al.* (2015), McKenzie *et al.* (2010), Marzec *et al.* (2014) and especially Edwards *et al.* (2008) give relatively detailed descriptions of their data cleaning and preparation. Wang *et al.* (2017) and Mukherjee and Chakraborty (2007) make no mention of the process used to obtain a document word matrix. Others like, Chen and Nayak (2007) and Devaney *et al.* (2005), discuss only the specific steps they felt were important making little to no mention of the rest.

The only author who specified a KDDM methodology is Edwards *et al.* (2008) who followed the CRISP-DM model which emphasises the importance of comprehensive documentation for the entire knowledge discovery process including preprocessing

[1] For those without an experimental component their proposed techniques are discussed rather than what they actually did.

and other implementation details often neglected in literature. The value of this methodology can be seen from the fact that theirs is the most repeatable study providing sufficient details of every step and the order in which they were used to enable replication. Others, such as Wang *et al.* (2017), provide interesting results but insufficient details to reproduce their experiment exactly.

The multitude of different tools and techniques discussed in the theory in Chapter 3 is also evident in these papers, with no definitive set recommended by all. Marzec *et al.* (2014), McKenzie *et al.* (2010) and Chen and Nayak (2007) all seemingly support the no-free-lunch theorem (Section 3.1) with Marzec *et al.* (2014) stating that while similarities in class can provide useful guidance, it cannot be assumed that what worked well for one problem will work well for another. While this highlights the value of multiple case studies in literature, it also emphasises the importance of experimenting on each particular dataset. McKenzie *et al.* (2010) confirm that it is imperative that the selection of these methods must be optimised for each dataset with Chen and Nayak (2007) adding that this can vary substantially between datasets.

The basic data cleaning methods used to prepare text for tokenisation is discussed in the following section followed by the more advanced preprocessing steps of POS tagging, stop-word removal, stemming, feature representation and feature selection. Next, the algorithms are discussed, followed by the evaluation methods.

#### 4.1.3.1 Data Cleaning

Edwards *et al.* (2008) started with an exploratory assessment of their data to identify the most pertinent DQ problems and guide the cleaning and preprocessing effort (as required by CRISP-DM). While it is assumed that the others did too, Edwards *et al.* (2008) is the only one that explicitly states and describes this process giving some insight to their selection of preprocessing transforms.

In particular they note that no information value was apparent in the use of punctuation or capitalisation and therefore transformed all text to lowercase, and replaced all punctuation (and non-alphabetic characters such as @, $ and &) with whitespace to be used as splitting character in the subsequent tokenisation process. They also concatenated words frequently negated by the word "no" so that e.g. "no damage" was replaced by the single term "nodamage" to retain negation in the tokenisation process where it would otherwise be lost (indistinguishable from the token "damage" in a sentence "there was damage" vs "there was no damage") (Edwards *et al.*, 2008).

Only a few other authors made specific reference to capitalisation. For McKenzie *et al.* (2010), case differentiation was not a factor since all of their data was fully capitalised. Uz-Zaman *et al.* (2015) also performed case-normalisation before tokenisation with the remaining Marzec *et al.* (2014), Chen and Nayak (2007), Wang *et al.* (2017), Devaney *et al.* (2005) and Mukherjee and Chakraborty (2007) making no mention of retaining or removing case information.

Similarly, none of Chen and Nayak (2007), Wang *et al.* (2017), Devaney *et al.* (2005) or Mukherjee and Chakraborty (2007) mention punctuation in their studies and it is unclear whether retained (unusual but not unheard of in TM) or removed and not mentioned. Like Edwards *et al.* (2008), Marzec *et al.* (2014) and Uz-Zaman *et al.* (2015) removed punctuation through the tokenisation process (either by replacing with whitespace or by using as splitting token which has the same result). McKenzie *et al.* (2010), however, used a modified tokenizer to retain apostrophes before the letters "s", "t" and "d" to prevent breaking up contractions such as "can't" into two tokens: "can" and "t" which becomes impossible to distinguish from its opposite "can" in the bag-of-words model. Due to the ubiquity of these steps, it is possible that the studies making no mention of punctuation or case differentiation did both but did not mention it.

Only Uz-Zaman *et al.* (2015) and McKenzie *et al.* (2010) addressed the numeric terms found in between the text of their dataset. Uz-Zaman *et al.* (2015) simply removed all number terms replacing them with whitespace before the tokenisation process. McKenzie *et al.* (2010) used several Regular Expressions to tag frequently reoccurring token-types such as codes (letter followed by numbers) and digits (all other number formats) to retain some information while substantially reducing the dimensionality of a feature space where every number is considered a separate feature.

An additional data cleaning step performed only by Marzec *et al.* (2014) is transliteration; unification of the way in which special characters such as äó are encoded which they did using a simple ASCII transformation (e.g. ä → a and ó → o) with great success. This is perhaps because their data is in Polish, which has a much higher degree of diaeresis than the English language considered by the other studies reviewed here.

Before this data cleaning process, Edwards *et al.* (2008), Marzec *et al.* (2014) and Devaney *et al.* (2005) all recommend improving the quality of data by utilising the Microsoft Excel spell-checker to flag incorrectly spelt words (typos, non-standard abbreviations and true misspellings) and manually correct these according to the researchers' best guess. This can have a drastic impact on the results considering the poor quality of data typically observed in maintenance records. However, this approach is not automated meaning any subsequent data will have to undergo the same manual effort before being analysed (test-performance is indicative of cleaned data performance and cannot be expected to perform similarly on new, uncleaned data). While automated spelling correctors have been proposed in other literature, Forman (2007) cautions that these do not work well on data with many technical terms (as in the maintenance domain) stating that mistakes can outweigh the benefits if words are incorrectly "corrected".

In the secondary studies, it is useful to mention that before removing punctuation and capitalisation, Rajpathak (2013) and Rajpathak and De (2016) create an abbreviation disambiguation algorithm to first identify and then standardise the large variation of abbreviations found in these records. They use a number of Regular Expressions to identify the most frequently occurring abbreviations using this case and punctuation information such as: <period, white-space, capital letter> = sentence boundary indicator, <period, white-space, lower-case letter> or <letter, period, letter> = abbreviation or acronym indicator. The disambiguation process is performed using their domain specific ontology to standardise the various forms of the same abbreviations into a single representative token.

In the primary studies Edwards *et al.* (2008) also standardise the various forms of the same abbreviation to a single term during their manual spell-checking process (e.g. "air con", "aircon", "ac", all replaced by one standardised: "aircondition") but this was not an automated process. While McKenzie *et al.* (2010) does not try to disambiguate abbreviations, they do prevent their splitting up by further modification on how their tokenizer handles punctuation to e.g. retain the abbreviated token "W/IN" (within) as is and not split it into "W", "IN".

#### 4.1.3.2 Stop Words and Stemming

All of Chen and Nayak (2007), Edwards *et al.* (2008), Marzec *et al.* (2014) and Devaney *et al.* (2005) performed both stemming and stop-word removal and Uz-Zaman *et al.* (2015) performed only stop-word removal. The remaining papers made no mention of these methods in their experimental designs. However, only McKenzie *et al.* (2010) gave sufficient detail of their preprocessing that this omission can be considered relevant. In Mukherjee and Chakraborty (2007) and Wang *et al.* (2017) it is unclear whether these steps were truly omitted from their preprocessing or simply not mentioned as they provide almost no detail of their preprocessing.

Chen and Nayak (2007) evaluated the effect of stop-word removal and stemming by treating it as a hyperparameter to compare the inclusion of both, neither, or either one alone on their clustering results. They report that the inclusion of both gave the best results and neither the worst although it should be mentioned that clustering performance is difficult to quantify. Their results also seem to indicate that stemming with no stop-word removal was preferable over stop-word removal with no stemming. They used the Porter stemmer. None of the other papers gave their reason for either the inclusion or exclusion of stop-word removal or stemming, nor evaluated its success.

#### 4.1.3.3 POS Tagging

The only authors to consider POS tagging were those concerned with information extraction rather classification, namely Mukherjee and Chakraborty (2007), Devaney *et al.* (2005) and McKenzie *et al.* (2010) of which only McKenzie *et al.* (2010) had an experimental component. Both Mukherjee and Chakraborty (2007) and McKenzie *et al.* (2010) state that the malformed, incomplete sentence fragments found in maintenance records are problematic for standard POS taggers (created for Standard English) which depend on the formulation of complete sentences. They therefore propose creating their own custom POS taggers.

Mukherjee and Chakraborty (2007) propose using the WordNet Knowledge Base (linguistic ontology), the Bill of Materials for the machine under consideration, and a set of expert provided domain rules to create a custom POS tagger. But since they had no data available, their method has not been verified experimentally.

McKenzie *et al.* (2010) propose training a supervised POS tagger on a labelled corpus (manually tagged with their respective parts of speech) and evaluates it on their sample of helicopter maintenance records. While they report tagging accuracies above 90% indicating the merit of their approach, they did not evaluate its impact as preprocessing method for subsequent classification tasks. Moreover, this approach would require double the labelling effort which is already problematic for maintenance datasets.

#### 4.1.3.4 Feature Representation

From their literature survey, Edwards *et al.* (2008) confirm the predominance of the Bag-Of-Words (BOW) approach. This is also evident from the studies reviewed here as all of the primary studies make use of the BOW model discarding the word-order and retaining only the frequency information of unigrams. Only McKenzie *et al.* (2010), who performed information extraction rather than classification, considered bigrams as well. While they report disappointing results for the bigrams compared to the unigrams, they ascribe this to their small sample of training data stating that bigrams require much more data than unigrams.

The most popular feature weighting scheme is term frequencies weighted by inverse document frequencies (TFIDF) which is used by Edwards *et al.* (2008), Marzec *et al.* (2014) and Chen and Nayak (2007). Uz-Zaman *et al.* (2015) uses term frequencies (TF) and Wang *et al.* (2017) does not specify which weighting scheme they used. While not evaluating another weighting scheme, Chen and Nayak (2007) reports that TFIDF did not work as well as expected for short documents. They speculate that this is because while important terms appear infrequently across the dataset, due to the small number of terms in every document, even important words are unlikely to occur frequently in any given document and are subsequently down-weighted by the low TF component in the TFIDF score. This suggests that binary features combined with an IDF weighting scheme may be a viable option for short documents but this was not investigated by any of these studies.

Edwards *et al.* (2008) and Marzec *et al.* (2014) further transform their TFIDF features to a lower dimensionality by computing the Singular Value Decomposition (SVD) of their Document Term Matrix. Marzec *et al.* (2014) combine the SVD features with their original TFIDF features into a single feature space from which the top features were selected. In Edwards *et al.* (2008) the SVD was only performed to reduce the dimensionality of the input used to train a Neural Network, with the TFIDF features used in a Decision Tree. While not evaluating these algorithms with the alternate feature representation, they report that the TFIDF features were superior to the SVD features stating that while the SVD features simplified the problem space, the information lost in this process reduced the performance. They also recognise the disadvantage of the loss of interpretability that accompanies SVD transformations (Edwards *et al.*, 2008).

#### 4.1.3.5 Feature Selection

All of the classification and clustering studies performed feature selection to some degree. Uz-Zaman *et al.* (2015) performed only maximum frequency thresholding to remove the most common words. Chen and Nayak (2007) performed both minimum and maximum frequency thresholding followed by selecting the top TFIDF-weighted features. They note that low thresholds should be used for minimum frequency thresholding because some important terms that reflect the failure mode do not occur frequently in the dataset (would be upweighted by IDF component in the subsequent TFIDF selection if not discarded). While not being able to do so due to lack of expert involvement, they state that SMEs should ideally adjust the thresholds until they get the desired results and manually discard any terms deemed irrelevant.

Edwards *et al.* (2008), who also had no SME involvement, inspected their top TFIDF-weighted features themselves to remove high-ranking but irrelevant terms before selecting the 100 highest weighted features. They state that this should ideally be done by SMEs.

Marzec *et al.* (2014) is the only primary study with intensive SME involvement. After minimum frequency thresholding, in which they removed all hapax legomena to retain only words appearing more than once (reduced feature set from 3000 to 206 words), domain experts selected the 76 words they deemed most relevant to the classification objective. These words were used to construct both the TFIDF and SVD features from which the top 15 word and SVD features were selected using the CHI-squared statistic.

Wang *et al.* (2017) note that feature selection is negatively affected by class imbalance as the features in minority classes tend to be dominated by features in the majority classes. They therefore propose a modified CHI-squared statistic that adjusts the feature weights of the minority and majority classes to make them further apart. They compare this with various other feature selection methods including the standard CHI-square statistic, Information Gain and the original feature set (no feature selection).

The original feature set (with no feature selection) is consistently worst showing that not all features are helpful to the algorithm. Both CHI-squared and Information Gain improved the performance with the standard CHI-squared performing marginally better than Information Gain and the best results achieved by their modified CHI statistic. It is interesting to note the consistently lowest performance by no feature selection given the fact that they used a Support Vector Machine, an algorithm which is commonly believed not to benefit from feature selection, and supports Forman (2004) who states that this is a popular myth (Wang *et al.*, 2017).

#### 4.1.3.6 Algorithms

The algorithms observed in the classification studies include Decision Trees (Marzec *et al.* 2014; Edwards *et al.* 2008; Bastos *et al.* 2014), Support Vector Machines (Wang *et al.* 2017; Sipos *et al.* 2014), Neural Networks (Edwards *et al.*, 2008) and Naïve Bayes (Uz-Zaman *et al.*, 2015).

Various reasons are given for the respective algorithm choices. Bastos *et al.* (2014) chose Decision Trees (DT) for its interpretability stating that in the maintenance domain, interpretable models are preferable to black-box models like Neural Networks (NN) which are popular in applications where only accuracy matters. Edwards *et al.* (2008) evaluated both Neural Network and Decision Tree algorithms. Like Bastos *et al.* (2014) they chose a Decision Tree for its interpretability, even stating that the accuracy was not expected to be high, but that they hoped it could provide some useful insights to the data for use in other models. The more complex Neural Network was selected for its anticipated superior accuracy but, contrary to their own expectations, they found that it was outperformed by the simpler DT. This result was also the motivating factor in Marzec *et al.*'s (2014) choice of Decision Tree citing Edwards *et al.*'s (2008) paper in their algorithm selection.

Sipos *et al.* (2014) chose a Support Vector Machine (SVM) with a linear kernel citing its proficiency in dealing with the high dimensionality and sparsity of text data. While they do not provide the results, they state that a K-Nearest Neighbours algorithm was also evaluated but performed very poorly on their dataset. They also evaluated building an ensemble of classifiers but report that while it increased the computational complexity, it did not noticeably improve the performance over the single SVM. Wang *et al.* (2017) does not give a reason for choosing Support Vector Machines but it is not a surprising choice since they are often regarded as best-in class for text problems. While Sipos *et al.* (2014) cautions that SVM may struggle with highly imbalanced datasets, according to Wang *et al.* (2017), this is true of all algorithms stating that all classifiers are inclined towards the majority class.

Uz-Zaman *et al.* (2015) selects the multinomial Naïve Bayes algorithm based on its proficiency at handling high-dimensional data such as text and its successful application in other text classification problems (such as email spam-filters). Only Edwards *et al.* (2008) compared two different algorithms while the rest evaluated only one.

#### 4.1.3.7 Evaluation

While it is interesting to compare the different preprocessing and modelling tools used in these studies, a more pertinent concern is their evaluation process as this concerns the validity of the results.

Only Sipos *et al.* (2014), which is considered a secondary study in this review, performed cross-validation to evaluate and optimise their models. None of the primary classification studies used cross-validation to evaluate their models. Only Wang *et al.* (2017) performed a three-way training, testing and validation split while the rest performed a simple two-way split (Marzec *et al.* 2014; Edwards *et al.* 2008; Uz-Zaman *et al.* 2015). The absence of cross-validation is noteworthy since it is widely preferred in the theoretical literature. While a three-way split is acceptable, a two-way split is susceptible to overfitting and may lead to overestimated results and is widely recommended against.

Uz-Zaman *et al.* (2015) makes no mention of the strategy used to split their data into training and testing sets. Both Edwards *et al.* (2008) and Marzec *et al.* (2014) used a random splitting strategy while Sipos *et al.* (2014) and Wang *et al.* (2017) used a stratified strategy. Neither Marzec *et al.* (2014), nor Wang *et al.* (2017) justified their respective choices, and while both Edwards *et al.* (2008) and Sipos *et al.* (2014) reported class imbalance in their respective datasets, according to Edwards *et al.* (2008) this was not extreme enough to require stratified sampling while according to Sipos *et al.* (2014) it was.

The other secondary study by Bastos *et al.* (2014) also performed a three-way evaluation, but they split the data chronologically using a time-series, sequential splitting strategy. Their dataset, spanning a year, is separated into three sequential sets with the first two used as training and validation sets (to optimise the model) and the chronologically last set used as hold-out test set to evaluate the model. In this way the model is only ever trained on "historical" data and evaluated on "future" data as would be the case in deployment.

The various evaluation metrics discussed in Section 3.7.4.1 are evident in these studies. In Chapter 3 it was discussed that reporting on the Accuracy[2] score in isolation can be misleading for data presenting class imbalance. While Edwards *et al.* (2008) did report class imbalance in their study, they provide only the accuracy (although in the form of its compliment misclassification error). Uz-Zaman *et al.* (2015), who also reports significant class-imbalance, provides the Precision and Recall in addition to the Accuracy and also evaluate the learning curves for increasing dataset sizes for each of these metrics. From their results it can be seen that while both the Precision and Recall follow the expected test curves (rapid increase followed by a plateau), the Accuracy follows a flat, horizontal line irrespective of the dataset size and seemingly more indicative of the class distribution than the actual model performance. At 80% the Accuracy is consistently significantly higher than both the Precision (40%) and the Recall (30%). Together these results show why Accuracy must not be considered in isolation.

Only Wang *et al.* (2017) and McKenzie *et al.* (2010) report the F-1 score to evaluate the performance of their classification and information extraction model respectively. While they do not specify how this was computed, the specific formulation is not that important since neither performed cross-validation (as per the study by Forman and Scholz (2010)). The only domain specific study that considered cross-validation was the secondary study by Sipos *et al.* (2014) who evaluated their models using the Area Under the Curve (AUC) metric.

In addition to the performance metrics, Sipos *et al.* (2014) and Edwards *et al.* (2008) also evaluated Dummy Estimators to determine how much of their performance can be attributed to chance for the particular datasets under consideration and serve as a baseline for their models. Sipos *et al.* (2014) reports outperforming a random baseline dummy estimator and Edwards *et al.* (2008) reports that three out of four models outperformed a majority class dummy estimator. Uz-Zaman *et al.* (2015) does not report a dummy baseline, but from the dataset characteristics and results provided it can be seen that the Precision and Recall would be outperformed by any of a majority class, random or stratified dummy estimator. (Their Accuracy score, on the other hand, would be outperformed by a major-

[2] As per Chapter 3, uncapitalized accuracy is used to refer to the general performance with regard to “correctness” while the capitalised Accuracy refers to the specific metric.

ity class dummy estimator, would surpass a random dummy estimator and be equivalent to a stratified dummy estimator once again showing the influence of class-imbalance on Accuracy.)

Since the decisions made with these models can have cost and safety implications, Marzec *et al.* (2014) emphasise the importance of achieving high accuracies. However, Edwards *et al.* (2008) note that even moderate accuracy can be acceptable for business decisions such as budgeting when compared to alternative intuition-based decision making.

### 4.1.4 Concerns

A notable concern observed in these studies is the pervasiveness of data leakage. This is due to researchers performing data cleaning and preprocessing transforms on the entire dataset (before separating the hold-out test) as well as repeated evaluations on the same test-set in a two-way split.

Bastos *et al.* (2014) and Uz-Zaman *et al.* (2015) explicitly state that the data preprocessing was performed before separating the hold-out test data. While not explicitly stated by Marzec *et al.* (2014) or Edwards *et al.* (2008), from the order in which their modelling steps are presented, it seems as if the same is true for their studies. This means that none of Bastos *et al.* (2014), Uz-Zaman *et al.* (2015), Marzec *et al.* (2014) or Edwards *et al.* (2008) truly performed a hold-out test evaluation since their models had unfair access to the test-set. For example, when tokenisation occurs before splitting, the models do not encounter any unseen words in the test sets, a very unlikely scenario in deployment.

Furthermore, evaluating the models using a two-way split (as done by Marzec *et al.* (2014), Edwards *et al.* (2008) and Uz-Zaman *et al.* (2015)) can also be considered as a form of data leakage since the model is repeatedly tested on the same dataset during the optimisation process and the parameter selection has unfair insight to the test data.

Data leakage can lead to overestimated results as the hold-out test set is not truly unseen creating the possibility of overfitting to that specific test set. While it is not clear how big the impact of this is on their results (may even be negligible for the specific datasets) it calls into question validity of their methodology.

Finally, none of the studies made explicit reference to the IID assumption, nor what its violation might mean for the results. However, chronological drift (which violates the IID assumption) is implicitly acknowledged by Mukherjee and Chakraborty (2007) who identify the *"changing dynamics of product life-cycles"* as a challenge in maintenance data; Rajpathak and De (2016) who refer to the changing failure rate of the "bathtub" curve, and by all the authors that recognise the heterogeneity of the data (Chen and Nayak 2007; Rajpathak *et al.* 2012; Devaney *et al.* 2005; Sipos *et al.* 2014).

The next section considers the broader text mining literature, focussing specifically on data with similar properties as the maintenance records, particularly those with short document lengths.

## 4.2 Domain Independent Literature

Beyond the maintenance domain, the most relevant studies are those concerned with Twitter data which share some of the characteristics of the maintenance records described above; in particular short document length and non-standard English usage. While the majority of text classification research has been focussed on corpora with documents longer than 100 words, the rise of social networking sites has made the classification of very short documents, called short-form corpora by Bermingham and Smeaton (2010), an important research topic (Timonen, 2012). Twitter allows people to publish their thoughts in real-time and therefore contains valuable information on the public sentiment regarding various social, political or economic issues (Mozetic *et al.* 2018; Mozetic *et al.* 2016; Bermingham and Smeaton 2010).

Until recently, Twitter messages were limited to 140 characters with an average length of 34 *characters* per tweet[3] (Perez, 2018) making it much shorter than more

[3] While the character limit has since been extended to 280 characters, according to Perez (2018) this has not had much of an effect on the average length of Tweets (which has actually

typical text-mining corpora such as the Reuters-21578 dataset with an average length of 160 *words* per record (Timonen, 2012). The character length limitations along with the conventions of social media has led to a high prevalence of non-standard English usage including emoticons, #hashtags, @handles, slang and informal abbreviations (such as "u", "btw" and "wtf") (Bermingham and Smeaton 2010; Mozetic *et al.* 2016). While these characteristics separate it from more traditional TM literature and resemble the challenges face by maintenance records, there are some important distinctions.

Due to social media already being in the public domain, these datasets are typically not proprietary and can be published for use and review by other researchers. This means the data labelling effort can be shared. Moreover, since the target audience for these messages are general society, the data can be effectively understood and labelled by lay-persons (including researchers) and do not require SMEs as in the maintenance domain. Finally, while the improper language usage separates Twitter from the more formal language found in typical TM corpora, the non-standard language is still relatively standardised across all of Twitter. (For example, unlike the maintenance domain where the abbreviations differ by organisations, departments and even individuals, Twitter abbreviations such as "U", "LOL" and "IMO" are widespread.)

While there are more Twitter classification studies available than can be evaluated here, it was thought useful to summarise a few of the most relevant ones found to show how similar problems are treated beyond the maintenance domain. The studies considered in this section includes that of Bermingham and Smeaton (2010), Timonen (2012), Mozetic *et al.* (2016), Mozetic *et al.* (2018) and Blamey *et al.* (2012). Thereafter, the scope was broadened beyond Twitter to verify the findings made here.

Bermingham and Smeaton (2010) consider the effect of document length on the performance and techniques used in sentiment classification by performing a number of experiments on two short-form corpora and their respective long-form coun-

since gone down to 33 characters) with only 12% longer than the original 140-character limit. Moreover, since all of the studies reviewed here used Tweets collected before this extension this is not a factor here.

terparts: micro-blogs (Twitter data), micro reviews, standard blogs and standard reviews. While they recognise the challenges posed by short-document length, they also recognise a possible advantage of such brevity stating that such texts are more focussed with less opportunity for non-relevant information to enter the content. They performed both multiclass and binary classification. Like Edwards *et al.* (2008), they consider the former more difficult than the latter and as they expected, report significantly higher results for the binary task.

While Bermingham and Smeaton (2010) found mixed results concerning the overall performance of short- over long-form corpora (microblogs outperformed blogs, but micro-reviews underperformed reviews), they report significant differences in the effectiveness of the various techniques. In particular they note that higher-order n-grams (bigrams and trigrams) did not improve the performance on short-form corpora; that binary features were more effective than frequency-based feature vectors; and that no benefit was gained from either stop-word removal or stemming. Interestingly they also found that while the SVM performance was superior for the long-form corpora, the opposite was true for the short-form corpora where it was outperformed by the Multinomial Naïve Bayes.

Timonen (2012) compares the performance of common feature weighting approaches on two corpora of very short documents consisting of Twitter and online consumer poll data respectively. According to Timonen (2012), the biggest challenge in short document classification concerns feature weighting. Because there are so few words per document, each word typically only occurs once per document, regardless of its importance. This makes traditional frequency-based feature weighting approaches such as TF and TFIDF ineffective on short documents as these are unable to distinguish between word-frequencies and end up distributing the weights somewhat equally among all words. They call this the *TF=1 challenge*. While they do not consider binary features in their experiments, their study provides some insight to the comparatively poor performance of frequency-based feature vectors in the studies by Bermingham and Smeaton (2010) and Chen and Nayak (2007).

Mozetic, et al. performed two sentiment classification studies on multiple corpora of Twitter data. In the first they consider the quality, quantity and sampling of training data as well as the choice of classification algorithm and evaluation metric (Mozetic *et al.*, 2016). In the second they address the evaluation procedure required to obtain reliable performance estimates and evaluate whether the temporal ordering of Twitter data matters (Mozetic *et al.*, 2018).

They recognise that human labelling depends on the subjective judgement of human annotators which may vary between different people and may even be inconsistent in individuals and recommend evaluating this using the annotator agreements described in Section 3.2. In particular they show the importance of sample selection to exclude low quality data and recommend removing these on the basis of annotator self-agreement. They find that quality is more important than quantity showing on their learning curves that while performance is typically improved by including more training data, if the quality of the additional data labels is too low (indicated by low self-agreement), it actually decreases the performance (Mozetic *et al.*, 2016).

Neither their literature review nor their experimental results found a significant difference between the performance of the top classification algorithms. They conclude that that the choice of algorithm is not highly significant and recommend that efforts should be spent on improving the quality of the training data rather than on selecting the best algorithm. They also consider different evaluation metrics and confirm that Accuracy is unacceptably misleading and recommend using the balanced F-score instead (Mozetic *et al.*, 2016).

To investigate whether the temporal nature of Twitter data matters, Mozetic *et al.* (2018) compare various performance evaluation methods that split the data in different ways. In particular, they evaluate several variants of cross-validation and sequential validation approaches. While Twitter data is time-ordered, it cannot be considered a time-series as Tweets are posted at any time and at any frequency. However, while original Tweets are not directly dependent on previous posts, long-term data drift is evident from changing trends in either the topics of interest or the sentiment regarding a certain topic (which can be affected by influential users or outside events) so that the data cannot be considered IID either (Mozetic *et al.*, 2018).

While sequential approaches provide more realistic test-scenarios where the training data always precedes the test data chronologically to emulate the deployment scenario, cross-validation utilises all the available data for both training and testing purposes leading to more robust performance estimates. However, they recognise that cross-validation is only applicable when the IID assumption holds, and that its violation, as in Twitter data, can hinder the generalisability of performance estimates. Along with the standard cross-validation approaches found in literature (random and stratified), they also evaluated Bergmeir and Benitez's (2012) grouped cross-validation method as an in-between approach that uses all of the data to yield more robust estimates, but slightly compensates for the IID violation (Mozetic *et al.*, 2018).

Their results support that of Bergmeir and Benitez (2012). They find that the cross-validation variants tend to overestimate the true model performance and the sequential methods tend to underestimate it. Randomly split cross-validation was the worst, and while recognising it as the industry standard, they state that it should not be used to evaluate time-ordered data such as this. While underestimation is usually preferable to overestimation, they do not find a notable difference between the best performing cross-validation (grouped CV) and sequential approaches and recommend using grouped CV for its added robustness (Mozetic *et al.*, 2018).

Blamey *et al.* (2012) also performed sentiment classification on Twitter data to evaluate the usefulness of character, rather than word, n-grams. They hoped that the increased flexibility of this approach would be well-suited to the non-standard English found in social media, but found little improvement over the standard BOW model. However, in contrast to Bermingham and Smeaton (2010), they found the inclusion of bigrams beneficial over unigrams alone (they did not evaluate trigrams).

### 4.2.1 General Observations

Unlike any of the domain specific studies discussed above, most of these authors published their data including that of Mozetic *et al.* (2016), Mozetic *et al.* (2018) and Bermingham and Smeaton (2010). While Timonen (2012) could not publish the data in its original text form, they did make their feature vectors publicly available. Moreover, according to Mozetic *et al.* (2016) there are already several publicly available, manually labelled Twitter datasets, ranging in size from several hundred to several thousand records. Blamey *et al.* (2012), who did not specify whether their dataset is published, also refers to Twitter-specific resources available such an emoticon tagger obtained from Wikipedia data. Furthermore, because Twitter data does not require SME labelling, labelled data was easier to obtain than for the domain specific studies. Both of Bermingham and Smeaton (2010) and Mozetic *et al.* (2016) were even able to address the quality of their data through extensive annotator training and duplicate labelling to evaluate the annotator-agreement measures.

As with the domain specific studies, there is a lot of variation in the specific preprocessing methods applied by each study, but the actual range of techniques considered is mostly consistent with the exception of document-length normalisation. While not mentioned by a single domain specific study, perhaps because it was thought irrelevant due to the short document lengths, it is used by both Bermingham and Smeaton (2010) and Timonen (2012) who also consider short-form corpora (both of which use L2 normalisation).

All of these studies evaluated at least two algorithms. This is notably different to the domain specific studies where only Edwards *et al.* (2008) evaluated more than one. Bermingham and Smeaton (2010) considers a Multinomial Naïve Bayes and a Support Vector Machine with a linear kernel stating that these were the state of the art in text classification. Interestingly, they found that while the SVM performance was superior for the long-form corpora, the opposite was true for the short-form corpora where it was outperformed by the Multinomial Naïve Bayes.

Timonen (2012), who considers only short-form corpora, evaluated three algorithms namely SVM, Naïve Bayes and KNN stating that all of these have shown good promise in text classification. They found that SVM was the clear winner followed by Naïve Bayes. While this contradicts the results of Bermingham and Smeaton (2010), it conforms to their own expectations as, like many in literature, Timonen (2012) considers SVM best-in class for text classification.

While noting the popularity of SVM, Mozetic *et al.* (2016) report that a wide range of algorithms are used in literature with apparently no consensus as to a best one, citing the contradictory results found in studies comparing them. They find no significant difference between the performance of the top classification algorithms in their literature review. Similarly, they find comparable results between multiple variants of SVM and Naïve Bayes in their experiments. From both their literature review and experimental results, they therefore conclude that the particular choice of algorithm is not highly significant.

Blamey *et al.* (2012) considers four algorithms (which includes SVM and Naïve Bayes) in their study finding mixed results with neither clearly superior. While not discussing the significance of this themselves, their results seem to support that of Mozetic *et al.* (2016).

In terms of model evaluation, all of these studies used cross-validation in comparison to none in the primary domain specific studies. However, only the studies by Mozitec, et al. address the implicit IID assumption made by cross validation and recognise its violation in the time-ordered Twitter data due to chronological drift. They therefore perform a grouped CV finding that other CV approaches, and especially randomised CV which is the industry standard, overestimates the true performance on time-ordered Twitter data in support of Bergmeir and Benitez (2012) (Mozetic *et al.* 2016; Mozetic *et al.* 2018). Neither Blamey *et al.* (2012) nor Bermingham and Smeaton (2010) specified the splitting strategy used and Timonen (2012) used a random splitting strategy.

Timonen (2012), Mozetic *et al.* (2016) and Mozetic *et al.* (2018) evaluate their models using the balanced F-score metric as recommended in Chapter 3, but do not specify how this was averaged across the folds of the cross-validation. While reporting on Accuracy in imbalanced data is widely recommended against (a notion supported by Mozetic *et al.* (2016)), both Bermingham and Smeaton (2010) and Blamey *et al.* (2012) report only on Accuracy. This is a greater concern for Blamey *et al.* (2012) who makes no mention of their class-distribution while Bermingham and Smeaton (2010) balanced their data artificially using under-sampling.

Some of the above points are also addressed in literature outside the Twitter domain. Leopold and Kindermann (2002), Ikonomakis *et al.* (2005) and Wilbur and Kim (2009) confirm the TF=1 challenge in short documents stating that due to the lack of variance in term frequencies, frequency-based feature vectors do not work as well on short-form corpora and recommend using Binary features in support of Timonen (2012) and Bermingham and Smeaton (2010). Ikonomakis *et al.* (2005) further recommend using Binary features on the basis of computational efficiency. Moreover, according to Wilbur and Kim (2009), the popular success of TFIDF comes from the IDF component and recommends discarding the TF and using IDF with Binary frequencies instead. It is interesting to note that while Wang and Manning (2012) found a slight advantage to using Binary features rather than TF features for long documents, they report a negligible difference in short documents.

Tan *et al.* (2002) support Bermingham and Smeaton (2010) who found that higher-order n-grams do not work well on short documents stating that phrases are even more unlikely to repeat in short documents aggravating the potential disadvantages of higher order n-grams (increased dimensionality, sparsity and synonymy). Bekkerman and Allan (2004) do not address document length in their study, but while they argue against the inclusion of bigrams in general, they do state that bigrams may be beneficial in technical domains with more limited corpus vocabularies (such as the maintenance domain) as informative phrases are more likely to repeat. As possible explanation for the contradictory results by Bermingham and Smeaton (2010) and Blamey *et al.* (2012), Wang and Manning (2012) found that the potential benefit of bigrams depends more on the specific task than on document length, reporting mixed results for the benefit of bigrams in both short and long form corpora.

Kobayashi *et al.* (2018) support Bermingham and Smeaton's (2010) findings that stop-word removal and stemming are not as effective for short documents. They also state that document-length normalisation does not affect short documents which may be why none of the domain specific studies mentioned it (Kobayashi *et al.*, 2018). However Leopold and Kindermann (2002) found that document-length normalisation did improve classification performance, even for short documents. But, in contrast to Bermingham and Smeaton (2010) and Timonen (2012) who applied L2 normalisation, as well as the general recommendation literature, Leopold and Kindermann (2002) found that for short documents L1 normalisation was sometimes superior with L2 only surpassing and becoming notably superior for increasing document lengths.

Finally, Wang and Manning (2012) support Bermingham and Smeaton's (2010) finding that Naïve Bayes performs better than Support Vector Machines on short form corpora while the opposite is true for longer documents. They speculate that this is due to the many poor assumptions of the Multinomial Naïve Bayes becoming increasingly detrimental for longer documents. They also state that while the Multinomial Naïve Bayes is generally considered superior to the Bernoulli variant, the performance becomes more comparable for short documents (Wang and Manning, 2012).

# Chapter 5

# Research Methodology

This chapter presents the overarching methodology according to which the empirical analysis was designed and executed. It starts with an overview of common Knowledge Discovery and Data Mining methodologies followed by a more detailed explanation of the one selected for this study, the Cross-Industry Standard Process for Data Mining (Crisp-DM). The more detailed experimental methodology is presented in the next chapter and was selected according to the principles of this methodology.

## 5.1 Knowledge Discovery and Data Mining Methodologies

As discussed in Chapter 2, this project falls within the realm of Knowledge Discovery and Data Mining (KDDM) and can therefore benefit from a Knowledge Discovery methodology. Several methodologies have been proposed (and used) in literature. These usually take the form of knowledge discovery process models that provide a roadmap for the planning and execution of a KDDM project (Sharma, 2008). Some of the most popular models include the nine-step model by Fayyad et al. (1996), the five-step model by Cabena et al. (1998), the eight-step model by Anand and Buchner (1998) and the six-step CRISP-DM model (1996).

Cios *et al.* (2007) and Kurgan and Musilek (2006) review the different models finding them very comparable. They all propose a multiple step, sequential process that spans a similar range of activities with the main difference lying in the number and scope of their specific steps (namely how the activities are grouped). They conclude that while there is no universally "best" methodology, each has its own strengths and weaknesses making it more or less preferable for different applications and recommend making a choice based on the application domain, user background and individual preferences among other things (Cios *et al.* 2007; Kurgan and Musilek 2006).

They further identify several common principles advocated by all of these models. The most important of this is the emphasis on iteration. While all these models outline a sequential approach with each new step building on the outputs of the previous; they all emphasize the need for multiple feedback loops contained in an iterative revision process. Another important point confirmed by all these methodologies, is the context-dependent application of these models which will vary according to the objectives, resources and complexity of each project. Finally, all of these methodologies recognise the data preparation step as the most time-consuming (and critically important) part of the knowledge discovery process; a fact that might be missed by industry practitioners as this step is seldomly documented in literature. (Cios *et al.* 2007; Kurgan and Musilek 2006)

Both Cios *et al.* (2007) and Kurgan and Musilek (2006) distinguish between methodologies developed by industry and those developed in academia. Similar to the academic-industry gap identified in ML literature (Section 1.2), Cios *et al.* (2007) and Kurgan and Musilek (2006) found that the academic tools do not take practical, industry issues into account. For this reason the CRISP-DM methodology (discussed below), which is the leading industry model according to both Cios *et al.* (2007) and Kurgan and Musilek (2006), was selected. Kurgan and Musilek (2006) further state that this methodology is the most appropriate model for novice practitioners, industry and academic alike, citing its easy-to-read documentation and intuitive, industry focussed descriptions. This methodology was also used by Edwards *et al.* (2008), the only domain specific paper found that stated their methodology.

## 5.2 CRISP-DM Model

The Cross-Industry Standard Process for Data Mining (CRISP-DM) was developed by a large group of industry professionals in the late 1990s (Chapman *et al.*, 2000). It falls within the delimitations set out in Section 1.4 as it was published as an open standard making it very accessible and contributing towards its popularity.

It identifies six main phases that should be completed in a KDDM project, namely Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. These can be seen in the reference model provided in Figure 5.1.

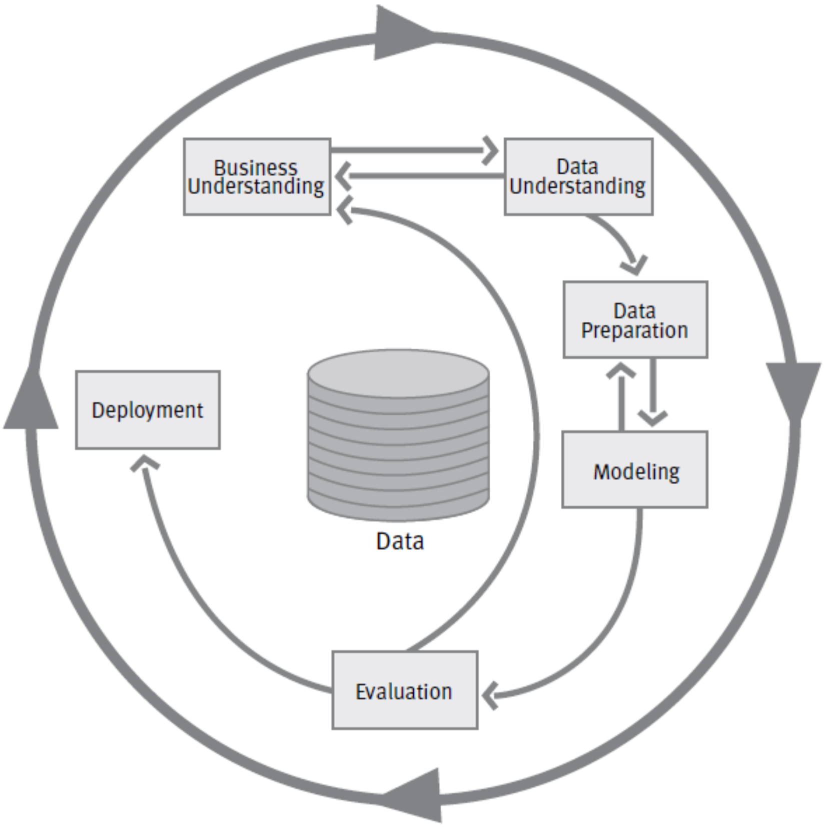


**Figure 5.1:** CRISP-DM reference model (Chapman *et al.*, 2000)

The model revolves around the data indicating the paramount importance of the nature, quality and quantity of this resource to the KDDM outcome. The circle indicates the cyclic nature of the process, KDDM is never really done and continues after the "solution" is found. The output of a KDDM project is often new information, either about the business or the data itself which enables new business questions, new processing strategies, collecting new data and essentially the start of a new (or more correctly continued) KDDM project (Chapman *et al.*, 2000). It is critical to realise then, that to continue receiving value from the data, the data mining activities must continue. This is in accordance with both ISO 55000 and ISO 8000 discussed in Chapter 2 which stresses the importance not only of establishing an information management system, but also the continued maintenance and management thereof.

Although the process model describes the six main phases in a sequential manner, the documentation stresses that this is for illustrative purposes only and is intended to show the general methodological progression, not a rigid rule-set. In practice there are various interdependencies between non-sequential phases of the methodology, the precise nature of which depends on the data-mining objectives, practitioner skill-level and resources as well as the specific dataset available. Like the other KDDM models, it acknowledges the iterative nature of the process and indicates main interactions with the multi-directional arrows (Cios *et al.* 2007; Kurgan and Musilek 2006; Chapman *et al.* 2000).

The first phase, *Business Understanding*, is paramount to the application of data mining. Any dataset contains a potential wealth-mine of information, but most of it is useless. Therefore the CRISP-DM methodology calls for the creation of both business and data mining objectives to focus the research effort. This is related to the requirement of *actionability* discussed in Section 3.7.4.2. It is important to get a good feel for the data that is available and the types of output information that is potentially valuable for the organisation (Chapman *et al.*, 2000). Due to the academic nature of this study, this is done in a more general manner for the asset management industry at large in Chapter 2. The research question formulated in Section 1.3 can be seen as the business objective for this more generic context with Chapter 6 providing only a brief overview of the organisational context within which the specific dataset was generated. The data mining objectives are presented in Section 6.3.

The second phase, *Data Understanding*, is about getting to know the available data through a combination of manual inspections, data queries, visualisation tools and statistical summaries. It is an exploratory process that will vary according to the particular dataset and research objectives, but will typically include general quality considerations (such as noise, error and duplication), as well as more specific data characteristics (such as interesting subsets of data and the distribution of key attributes). An important aspect of this phase is to establish the extent to which the various assumptions, such as IID, hold for the dataset as it can impact not only the performance but also the validity of the results. At this point the researcher will probably develop a better understanding of the business as well leading to more focussed research questions and objectives as indicated by the first multi-directional arrow in Figure 5.1 (Chapman *et al.*, 2000). The Data Understanding phase is not typically documented in literature because it is data and application specific holding little general value. For the same reason, only the most important outcomes of this phase are presented in Chapter 6. However, it is of utmost importance as it forms the basis of the subsequent Data Preparation, Modelling and Evaluation decisions and can have a profound impact on the final performance.

The third phase is *Data Preparation*. This typically forms the bulk (up to 60%) of the data mining effort and time (Cios *et al.* 2007; Kurgan and Musilek 2006). While it has no tangible business outputs, it is of critical importance as it involves the selection of features which is the only input given to the models. If important features are neglected or bad features are included it could drastically affect the ability of models to learn and may prevent the accomplishment of the business objectives. Some of the major steps identified by CRISP-DM is the selection, cleaning and formatting of data to the appropriate representation (Chapman *et al.*, 2000). These steps fall under the blanket term of preprocessing as defined in Chapter 3.

While CRISP-DM emphasises the importance of comprehensive documentation for all steps of the knowledge discovery process, preprocessing and other implementation details are often neglected in literature, considered too trivial for publication (Blamey *et al.*, 2012). This is not only an issue from an academic perspective (preventing replication), but can also be problematic for effective knowledge sharing within an organisation. For this reason both the selection and the evaluation of the preprocessing steps are documented in Chapters 6 and 7.

The fourth phase is *Modelling* and is where the actual model-building occurs. Typically, there are multiple algorithms that can solve a problem with no way to know beforehand which will be best as per the no-free-lunch theorem discussed previously. Furthermore, each algorithm also has multiple hyperparameters that can be adjusted including the selection of preprocessing techniques with which to combine it. The complex interactions between the different preprocessing steps and algorithms are acknowledged, but poorly understood in literature leading to either one-size-fits all preprocessing (discouraged by many authors such as Wang and Manning (2012) and Baharudin *et al.* (2010)) or brute force and random search methods to select optimal combinations from some specified set.

Several authors argue that preprocessing should be considered part of the modelling phase and be included in the hyperparameter optimisation to ensure not only the optimality but also the validity of the results (Krstajic *et al.* 2014; Tsamardinos *et al.* 2015; Varma and Simon 2006). This view is not incompatible with the CRISP-DM methodology which stresses not only the interrelatedness of all phases, but also the adaptability of their model (Chapman *et al.*, 2000). Furthermore, the inclusion of preprocessing in the model optimisation can simply be viewed as multiple successive feedback loops between the two phases in accordance with the multi-directional arrows between Data Preparation and Modelling in Figure 5.1. This is the approach used for the experimental analysis which is discussed in Chapter 6.

While the implementation details such as algorithm choice, optimisation strategy and evaluation scheme are beyond the scope of CRISP-DM (which strives to provide an application independent methodological guide) it recommends selecting and testing a few viable algorithms based on the specific KDDM objectives and what is observed in literature. Included in this phase is preliminary model assessment but this relates to model selection rather than evaluation of the results. It again emphasises the importance of adequate documentation for all steps of the analysis (Chapman *et al.*, 2000).

The next phase is *Evaluation* which not only seeks to assess the model performance, but also to understand the results (e.g. why certain techniques or parameter settings led to good/bad performance). While the predictive accuracy of the models are important, so are the other considerations identified in Section 3.7.4.2, namely efficiency, interpretability and actionability. These results are discussed in Chapter 7. Even more important, this phase also includes a methodological evaluation that reviews the experimental procedure in order to asses the validity of the results (Chapman *et al.*, 2000). This was identified as particularly important in the problem statement in Section 1.2 as business decisions depend on the reliability of these results which depend on the validity of the methods used to obtain them (Kobayashi *et al.*, 2018). This is provided in Chapter 8.

Included in the Evaluation phase, is determining whether the business objectives, as opposed to the data mining objectives, were met (Chapman *et al.*, 2000). For this study it involves answering the research question which is done in Chapter 9. The arrow pointing back to the first phase is not only indicative of the continuous nature of KDDM projects, but also shows that the outcome of such a cycle is enhanced Business Understanding that can be used to initiate new projects. This takes the form of the future recommendations also provided in Chapter 9.

The final phase of the CRISP-DM methodology is *Deployment*. Deployment refers to the application of the model to actually solve or improve the business problem (Chapman *et al.*, 2000). The most important outcomes of this phase concern the implementation of the model; the monitoring and maintenance of the model; and most important to this study, the documentation of the project which takes the form of the total thesis document. The research, models and results are provided to the data sponsor to use at their own discretion, but due to the academic nature of this project, the final deployment step is not included. This is a common limitation in literature as can be seen from the domain specific studies evaluated in Section 4.1 where only Rajpathak (2013) reports on the results of an industry deployed system.

# Chapter 6

# Empirical Analysis: Design

The empirical study was designed according to the CRISP-DM methodology discussed in the previous chapter and is presented according to the main phases of the process model shown in Figure 5.1. Although these phases are presented in a sequential manner, this is for the sake of clarity and presents only the final output of numerous iterations with a significant amount of back-and-forth between non-sequential phases (in accordance with the methodology).

It starts with the first phase of the CRISP-DM model, Business Understanding. Due to the academic nature of this study, it was considered appropriate to rather develop a more generic, domain understanding of the asset management industry (Chapter 2) instead of focussing on a particular business context. The research question in Section 1.3 is framed within this broad context and can be considered the business objective of the analysis. The section below provides only a brief overview of the specific organisational context within which the data was generated.

This is followed by the most important outcomes of the Data Understanding phase. The output of this phase was significantly more than is presented here, but holds little general value as it is dataset dependent. Hence, only the outcomes which are relevant to the remainder of the study is provided. This section is combined with the data selection which actually forms part of the Data Preparation phase, but is so integrated with the outcomes from the Data Understanding that it is easier to explain their combined output.

To ensure the validity of the results (in accordance with the eighth project objective), the Data Preparation, Modelling and Evaluation phases were integrated into a single procedure described in the section titled Experimental Design. Only the data selection and a few unsupervised (namely class independent) preprocessing steps were completed separately to lessen the computational load of unnecessary repetitions as recomended by Krstajic *et al.* (2014). These are described in Section 6.2.2.

## 6.1 Business Understanding

For the purposes of this study, maintenance records were obtained from one of South Africa's leading service fuel service-station brands. Effective management of these assets are critical, not only for business profitability, but also for environmental, health and safety considerations. Filling stations store and sell highly flammable and potentially explosive liquids that are also hazardous to the environment if not managed properly. Furthermore, many stations require 24/7 operability to meet customer expectations making unplanned downtime especially problematic.

The data was made available by a third-party maintenance and asset care provider with permission from their client. The maintenance service provider is responsible for keeping the service stations operational which includes breakdown response, routine inspections and various scheduled maintenance activities. The dataset contains historical work-order records detailing the maintenance events from more that ten years and have been stored in an SQL database.

The work-orders are created in an operational setting and may have many contributors including the maintenance technicians, call-centre staff and external contractors. While they are mostly used for scheduling and record-keeping purposes, they contain valuable information that could provide great benefit if utilised effectively. In an attempt to address this, numerous structured fields containing coded inputs were introduced to facilitate the computerised extraction of higher level information such as failure modes. For this they identified eight failure modes of interest and assigned them input codes. However, the completion rate has been poor with only the free-text fields consistently filled in.

## 6.2 Data Understanding, Selection and Cleaning

The total dataset contains 373 344 records spanning 12 years from 2005-2016 of which only 173 331 (46.43%) are labelled with a failure mode. Each record has a unique identifier code and represents a work-order generated for some maintenance event. The work-orders have several structured and unstructured fields that detail different aspects of the various events. This includes two unstructured, descriptive text fields detailing the *WorkRequired* and the *WorkPerformed* which are the possible inputs for the models. Several structured fields provide further details using coded inputs such as *SiteId*, *AssetId*, *WorkOrderStatus* and of course the field of interest, *FailureTypeId* which specifies the various failure modes (FM) and is the desired output of the models.

There are eight FM codes present in the dataset that correspond to high level functionalities such as *Electrical*, *Mechanical* or *Hydraulic Failure* which are shared by all asset types. This includes a non-informative *Generic Failure* class which makes up the majority at more than 80% of the labelled records. The two smallest classes make up less than 1% of the labelled set with the smallest containing only one record. This makes it a multi-class classification problem with severe class imbalance.

Records are initiated with the *WorkRequired* field making it a natural choice for model input as it is therefore present in all records. In the total dataset, this field varies in length from 1-2535 characters and 1-388 words. The average document length is 17.9 words with 251 single-word documents, 20 111 below five words, and 130 886 below ten. Without any preprocessing the corpus vocabulary is 102 425 unique words (of a total of 6 730 423 words). Included in this is unique numbers and words differing only by capitalisation. If all numbers, punctuation and accents are removed and the text forced to lower-case, this reduces substantially to 34 602 unique words. These properties are significantly different to those typically found in literature. For instance, the commonly used Reuters (RCV1-v2) dataset with 23 307 documents has a corpus vocabulary of 47 219 unique words and an average document length of 123.9 terms (Lewis *et al.*, 2004).

For this reason it was initially decided to combine the *WorkRequired* and *WorkPerformed* fields into a single text input to achieve longer document length. A preliminary analysis yielded suspiciously high results, and upon closer inspection it was found to be due to data-contamination (Section 3.8.3). At some point in time, a data-copy error occurred in the original database causing the text of all fields (including the target column *FailureTypeId*) to be copied into the *WorkPerformed* text field.

This is a classic example of data-contamination which causes an estimator to treat the duplicated label in the input text field as a strong indicator of that label-class. This creates a trivial classifier that is worthless for unlabelled records not containing this duplication. It can be difficult to identify this problem, especially when the contamination is not present in all records, and can lead to dangerously over-confident results. This is one of the reasons why the interpretability and not only the accuracy of the models are important as it is much easier to identify suspect logic in white-box models such as Naïve Bayes than in black-box models such as Neural Networks. For this reason, only the *WorkRequired* fields were used as modelling inputs for the analysis described in Section 6.4. This has the additional benefit of potentially being able to assist AM activities in real-time, namely before any work is performed.

A further challenge of the dataset, is the highly specific, non-standard vocabulary. Of the ten words with the highest frequency in standard English (*the, be, to, of, and, a, in, that, have, I* (Press, 2011)) only *the* appears in the top ten of the dataset, and is preceded by *pump*, a typically much scarcer word. Furthermore, there is a notable difference in the frequency of function (or stop) words in the dataset with specific words such as *dispensing*, *nozzle* and abbreviations such as *lrp* and *vp* included in the top ten which contains only three function words (most frequent of which is *not*). This indicates not only the domain specific vocabulary, but also the use of non-standard grammar as reported by McKenzie *et al.* (2010) among others with the text containing short phrases rather than well-formed sentences.

As evident from the domain specific studies evaluated in Section 4.1, one of the particular challenges in the analysis of maintenance records is the high variety of extremely terse, ad-hoc abbreviations used inconsistently throughout the dataset. To illustrate the quality difference between maintenance records and typical NLP corpora, Rajpathak (2013) compares the distribution of unique abbreviations in his dataset of automotive maintenance records with that of the Wall Street Journal corpus (a more general language dataset). He reports that only 6 different abbreviations account for 80% of all those found in the generic WSJ corpus in contrast with 107 in his dataset.

A preliminary inspection of the data seemed to indicate a similar trend in the data used for this study. Using simple heuristics based on observed punctuation and capitalisation trends, the frequency distribution of unique abbreviations was evaluated for the dataset finding that 85 abbreviations account for approximately 80% of the total. Although less than Rajpathak's (2013) dataset, this still indicates a variety of abbreviations that far exceeds that of the generic corpus. This is further confirmed by the inclusion of three abbreviations in the top ten most frequent words in the dataset, which is not the case in any of the generic corpora.

Apart from being non-standard, the abbreviations are also used inconsistently with many variants such as: *u.c*, *u/c*, *U/C*, *u/canopy*, *u-canopy*, *u canopy*, *u'c* and *under c* all referring to *under canopy* causing different versions of the same word to be treated separately. Not only does this inflate the already high dimensionality of text, but it can also the hurt performance as the feature significance is dissipated between different representations. All of these issues further restrict the effectiveness of standard NLP tools such as stemming and POS tagging that have been developed for standard English (McKenzie *et al.* 2010; Uz-Zaman *et al.* 2015).

### 6.2.1 Data Selection

The quality of the dataset is far from desirable and presents all the quality issues typically found in industry data namely incompleteness, duplication and noise. On top of the already significant challenges faced by standard NLP applications (high-dimensionality, sparsity, polysemy and synonymy), it also presents the domain specific challenges described in literature such as short document length, highly specific vocabulary, non-standard grammar, high number of extremely terse and inconsistently used abbreviations and most importantly, poor annotation quality. While machine learning is frequently applied to datasets that do not meet the normal standard of data quality (Edwards *et al.*, 2008), this does not imply immunity to the effects of poor data quality. In fact, Mozetic *et al.* (2016) demonstrates that data quality is the most significant determiner of model performance.

Mozetic *et al.* (2016) show that the annotation quality provides the upper limit of performance that can be reached and recommends removing low quality data on the basis of self-annotator agreement (Section 3.2). Unfortunately, annotator information was not retained and none of the documents were labelled more than once preventing sample selection on this basis. Instead, the data was filtered according to several quality issues identified during the Data Understanding phase.

The final sample comprised only a small portion of the total dataset containing 30 751 records (8.23% of the total). It was compiled according to the following steps discussed in more detail below:

1. Discard all unlabelled records (53.57% of total)
2. Discard all records outside 2007-2011 time period (65.63% of total)
3. Discard all uninformative classes (37.99%)
4. Discard all records that are not failure based (26.82% of total)
5. Discard all incomplete (cancelled, pending or rejected) records (43.53% of total)
6. Discard all externally completed, sub-contracted records (7.68% of total)
7. Discard all duplicate records (4.86% of total)
8. Discard all meta fields (preserving input, group and output)
9. Encode class labels

10. Shuffle data

The project was limited to supervised learning in Section 1.4 which requires labelled training data. Since only 173 331 (46.43%) of the records are labelled with a failure mode, the remaining unlabelled majority is excluded from use (step 1). While possible to expand the labelled set using a best-guess approach as done by Edwards *et al.* (2008), it was decided against this as the non-expert labelling would be inferior to (and inconsistent with) the subject matter expert (SME) labelling which could potentially decrease the quality of the training set. Furthermore, because the final dataset size is comparable to (and larger than many of) the related studies evaluated in Section 4.1, the additional time and effort investment was not judged worthwhile.

The yearly class distribution of the labelled sample is shown in Figure 6.1 from which it is clear that the independent, identically distributed (IID) assumption does not hold. There is evidence of both data-fracture and chronological concept-drift in the data (Section 3.1). From 2012 onwards there is a sudden onset of *Generic* labels accompanied by a drop for all the other classes. A similar discontinuity between records from before 2012 and after was observed for many of the other structured fields indicating 2011-2012 as a point of data-fracture due to changes in the information management system (evident from distinct code-sets for some fields). Such an extreme fracture point cannot be modelled as the relationship between the input and output, namely what constitutes as *Generic Failure* in terms of the input text, has emphatically changed (irrespective of whether this change is a reflection of the external reality).

For this reason only records from the period 2007-2011 were retained (the dataset starts in 2005, but the labelling only started in 2007) (step 2). While using the most recent data would be preferable, the labels become effectively meaningless after 2011 as all the documents are labelled with the same, uninformative class: *Generic Failure* making the data before the fracture point more valuable.

While more homogeneous, this sample is not IID either as can be seen by the yearly class distribution of the final sample shown in Figure 6.2 which still presents

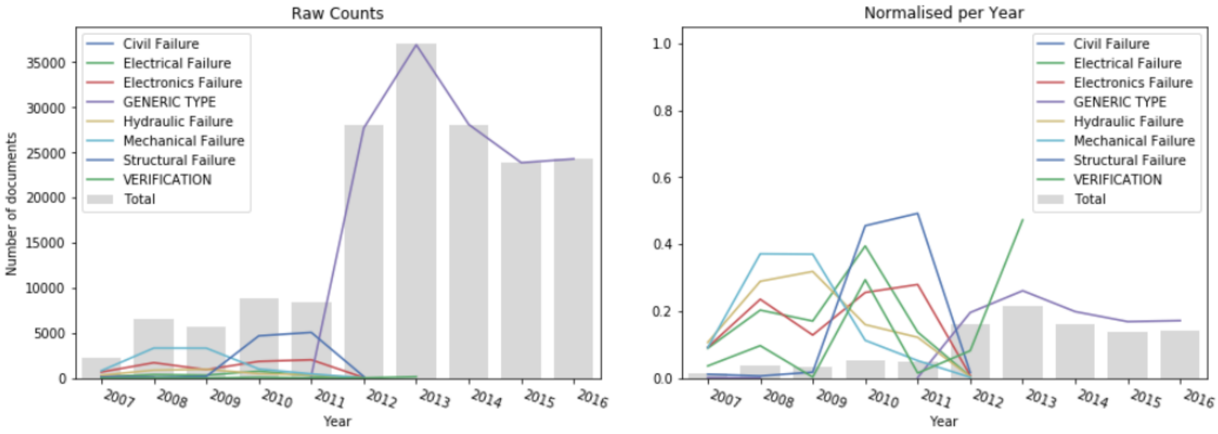


**Figure 6.1:** The yearly class distribution for the full sample showing the IID violation and a fracture point at 2011/2012

changes with time. The 2007-2009 and 2011-2012 subsets are more homogeneous than their combination but the difference is less extreme affecting only the *Mechanical* and *Structural* classes significantly. This indicates chronological data-drift, rather than data-fracture, which is a reflection of the non-stationary business environment (e.g. changing reliability curves of ageing assets or maturing AM practices leading to different types of failure). While none of the domain specific studies explicitly address the IID assumption, its violation and chronological drift is implicitly acknowledged by Mukherjee and Chakraborty (2007) who identify the *"changing dynamics of product life-cycles"* as a challenge in maintenance data, Rajpathak and De (2016) who refer to the changing failure rate of the "bathtub" curve and by all the authors that recognise the heterogeneity of the data (Chen and Nayak 2007; Rajpathak *et al.* 2012; Devaney *et al.* 2005; Sipos *et al.* 2014).

The sample was not reduced to either of the more homogeneous subsets for two reasons. Firstly, to maintain a larger quantity of training data, and more importantly, to accurately reflect the reality of the changing operating environment in which the model will be deployed to ensure the validity of the results. This was done by identifying the record date (year) as a group variable which is used to test the out-of-sample performance of the models in the evaluation procedure described in Section 6.4 below.

Both the largest and the two smallest classes were discarded for being uninformative leaving the sample with five remaining labels (step 3). The largest, *Generic Failure*, does not contain any information about the failure mode other than the absence of the others. This makes it a difficult class to model as it is not a homogeneous group. The smallest two classes, *Civil Failure* and *Verification*, contain 1 and 330 documents respectively. One example is obviously inadequate and while 330 documents are more plausible, it becomes very small when separated into the respective training and evaluation folds for the nested cross-validation (described in Section 6.4). Furthermore, like the *Generic* class, *Verification* does not present an actual failure mode but serves as a place-holder for activities such as routine inspection and machinery calibration. This reduces the class-imbalance somewhat with *Structural Failure* being the new majority class at 32% and *Electrical Failure* the smallest at 6%.

Because the modelling objective is the identification of failure modes, maintenance events that involved no failure was deemed irrelevant to the analysis. These records were eliminated by discarding all system-generated records (mostly scheduled inspections) and retaining only unscheduled, non-tactical records (step 4).

All unfinished records (indicated as cancelled, pending or rejected in the *WorkOrderStatus*) were discarded as the information of these records have not been verified as legitimate for whatever reason (step 5). Many of the reasons make the records irrelevant to the analysis such as false alarms indicating that no failure occurred, or out-of-scope statements absolving the service provider of responsibility. Other reasons, such as administrative or procedural errors, could indicate bad quality information which could hurt the performance. The same logic was followed in the decision to exclude all externally completed, sub-contracted records (indicated as company-owned in the asset-type). Since these assets fall outside the scope of the service provider's responsibility, the failure modes and maintenance of these assets are also considered irrelevant as well as potentially incompatible with the in-scope assets (step 6).

Next, the meta fields were used to identify and discard duplicate records (step 7) as it became evident during the data exploration process that some records with identical input and output fields nonetheless account for separate, albeit similar, events. To distinguish these from true duplicates (due to data handling errors) only those records with identical meta fields (*AssetID*, *SiteID* and *AuthorID*) were discarded. Thereafter, all meta fields were discarded (step 8) retaining only the input: *WorkRequired*, the group: *ReceivedOn* and output: *FailureModes*. Finally, the class labels were encoded to numeric format using Scikit-learn LabelEncoder (step 9) and the final dataset shuffled (step 10).

The final sample is much smaller than the original dataset containing 30 751 records (8.23% of the total). However, the quality is significantly better. The *WorkRequired* field of this sample has no more single-word documents and the document length is more tightly distributed than before at 15-563 characters and 2-93 words. The average document length has reduced to 10.7 words due to the exclusion of several extremely long, noisy outliers in the full set.

### 6.2.2 Unsupervised Data Cleaning

Although common in practice, many authors caution against preprocessing data prior to the cross-validation. The selection of preprocessing transforms amount to an optimisation process which can lead to biased results if adjusted repeatedly to improve performance. Just like the estimator can over-fit to the training data, so the modelling process can over-fit to the evaluation data leading to an over-estimated generalisation performance. Furthermore, care must be taken to limit the pre-processing transforms to the training data (hence be applied per training fold of a cross-validation). Data-leakage can occur if the full data-set is preprocessed before evaluation, as the model will not be tested on any unseen words and feature selection and weighting algorithms will have unfair access to feature distribution in the test set. This can also lead to optimistically biased results. However, Krstajic *et al.* (2014) notes that unsupervised (namely class independent) preprocessing procedures can be applied prior to the cross-validation if care is taken to prevent data-leakage in order to lessen the computational expense of unnecessary repetitions.

Following the recommendations by these authors, only a very limited degree of unsupervised preprocessing was performed prior to the evaluation comprising of data cleaning activities which falls under the Data Preparation phase of the CRISP-DM model. A small amount of data-leakage can still occur from the selection of unsupervised data cleaning transforms if the practitioner is guided too extensively by the particular elements of the training set. To prevent this, the exploratory analysis that is needed to guide the selection of cleaning steps was performed on the *WorkRequired* fields of the out-of-sample records which were excluded from the training set for various reasons in the above section. This way it is still a data-guided process but cannot over-fit to the particular training set. The remainder of the data preparation was performed in the inner loop of the nested cross-validation treating the selection of preprocessing variables as hyperparameters. These were optimised along with the estimator parameters according to the hyperparameter optimisation strategy described in Section 6.4.

Data cleaning is used to improve the quality and lessen the excessive dimensionality of the typically noisy text data. The following steps were applied in the order provided with more details below:

1. Transform all characters to ASCII equivalents (transliteration)
2. Replace all explicit newline characters: \n with single whitespace
3. Replace all numeric terms with a single numeric placeholder surrounded by whitespace: “ 0 ”
4. Abbreviation Standardisation using a heuristic search-and-replace function
5. Punctuation
    - Remove all apostrophes without replacement
    - Replace all remaining punctuation with a single whitespace
6. Remove capitalisation

The transliteration was performed using Python’s Unidecode module to replace inconsistently used accented characters such as *é* and *ç* with ASCII equivalents *e*, *c* to prevent inconsistent variants of words such as *facade* and *façade* from artificially inflating the dimensionality (step 1). Marzec *et al.* (2014) reports good results using this technique on a Polish dataset – a language with a much higher prevalence of diacritic marks)

Furthermore, the module is capable of intelligent replacement of special characters such as ° with a text equivalent *deg* while all remaining unknown character such as □ and � (remnants from earlier file conversions and corruption) are removed without replacement. In addition to the unknown character symbols removed by the transliteration, previous data handling errors also left some records with explicit newline characters: \n. These were replaced with a single whitespace (step 2).

The inclusion of numeric terms in the text data is very much context dependent based on whether or not the practitioner deems them relevant to the learning objective. If uncertain, Kobayashi *et al.* (2018) recommends retaining them and letting subsequent dimensionality reduction procedures such as feature selection handle their inclusion. However, due to the technical nature of the dataset, the unique numeric terms account for a disproportional portion of the dimensionality. Of the 102 425 unique terms in the unprocessed total dataset, 44 164 are numeric accounting for 43.12% of the dimensionality. Therefore, it was decided to merge all numeric terms into a single template, replacing them with a single numeric placeholder surrounded by whitespace namely “ 0 ” so that the numeric identity is retained without the excessive dimensionality of the specific values. (A method supported by Kobayashi *et al.* (2018) and McKenzie *et al.* (2010)) (step 3).

Many authors recommend cleaning the data by removing all non-alphabetic characters (step 5) and converting the remaining text into lower-case (step 6). This is typically done in the tokenisation process to remove extraneous differences between features and in so doing reduce the dimensionality of the input space. Rather than doing this in one fell swoop, it was decided to do this in stages enabling finer control of the tokenisation process as well as enabling the use of syntactic information (punctuation and capitalisation) in the abbreviation standardisation process (step 4) described below.

The standard way of handling punctuation is to treat it as a delimiter in the tokenisation process so that the text is split on (and removes) all whitespace and punctuation. However, as McKenzie *et al.* (2010) points out, this can lead to separating contractions and abbreviations in an undesirable way. For example [can't] is separated into [can] [t] preventing distinction between it and its opposite [can]. Abbreviations such as [e.g.] is separated into individual letters [e] [g] making it lose all meaning considering that word order is lost (and will return a shared feature [e] for very different abbreviations (such as e.g. i.e. and e.t.a) Rather than altering the tokenisation process like McKenzie *et al.* (2010), the punctuation marks that should not be separated on was identified and removed (without replacement) beforehand so that only valid delimiters remained.

In the first step, all apostrophes were removed without replacement reducing contractions like [can't] into a single token [cant]. This was preferred to McKenzie *et al.*'s (2010) method of ignoring apostrophes in the tokenisation process (yielding a final token [can't]) because it was observed that apostrophes were sometimes neglected in the text which would yield two different tokens.

In the next step, the punctuation elements that frequently occur in abbreviations (and should not be spilt on) were identified as periods, hyphens, and forward slashes. These were used in the abbreviation standardisation process described below. However, these elements do not only occur in abbreviations. Unlike the apostrophes, these elements are sometimes needed as delimitators and cannot simply be removed (or ignored in the tokenisation process). For example when whitespace is neglected after a sentence ending period or when a forward slash is used to separate items in a list, simply removing these symbols will lead to the concatenation of separate features leading to loss of the true features as well as addition of bad features (e.g. [switch/wiring] will lead to addition of bad feature: [switchwiring] that is very rare, and loss of both [switch] and [wiring] which is more common). For this reason, all abbreviations are standardised to punctuation free forms so that the remaining periods, hyphens, and forward slashes can be treated as delimitators.

Finally, the capitalisation trends relating to abbreviations were identified and used in the abbreviation standardisation process to identify punctuation-free abbreviations as well as helping to distinguish between punctuation elements used in abbreviations and without. After identification and lower-case standardisation of abbreviations such as *P* and *T* to *pump* and *tank*, the remaining text was converted to lower case.

Abbreviations are relatively rare in general-purpose text and are therefore not a big consideration in typical text analyses. However, as discussed in Section 6.2, this dataset contains both a high density and a high variety of non-standard abbreviations. This is a typical characteristic of maintenance records and one of the particular challenges that set it apart from more generic text analyses.

Abbreviations were identified in an exploratory manner using several Regular Expressions (regex)[1] based on the syntactic structure observed in the text. Because a significant portion of these were found to be variations of each other, further expressions were constructed to search for additional variants of the most frequent abbreviations and combining those found into a single abbreviation class and list of corresponding regex expressions. For each abbreviation class, a standardised version was selected and the regex rules of all variants combined into a single search-and-replace function.

To enable the subsequent removal of all remaining punctuation and capitalisation, the abbreviations were all standardised to lower case, punctuation free formats containing only delimiter whitespace. The standardised versions were selected on a per-case basis. For some abbreviations such as *u/c*, the intended meaning (under canopy) was obvious and consistent. For these, the unabbreviated text form was used as the standardised replacement to retain the relationship between unabbreviated occurrences of the phrase as well as individual occurrences of either term. In this manner the relationship between records referring to e.g. *under canopy*, *all canopy* and *side canopy* is maintained by the shared feature: *canopy* which would be lost with abbreviated replacements (*uc*, *all canopy* and *sc*). However, for some abbreviations the intended meaning could not be inferred from the text, while for others such as *d/b*, multiple context-dependent meanings (e.g. *data base*,

[1] A sequence of characters that defines a search pattern and is more powerful than exact match searches. For example: re.search(r'\b[\w]/[\w]+', flags=re.IGNORECASE) will search for all occurrences of a single letter followed by a forward slash followed by one or more letters to match both u/canopy and U/C but not switch/wiring

*distribution board*, *dash board*) were identified. These were standardised to a set abbreviated format: lower case, no punctuation and no spacing e.g. *db*. Although still ambiguous, the standardised format prevents splitting abbreviations in the tokenisation process and reduces the dimensionality of the input space by the removing extraneous differences in their formulation.

To prevent data contamination, this procedure was conducted on records excluded from the final sample in Section 6.2. Only the final search-and-replace function is applied to the actual dataset. This ensures that the cross-validation test sets remain truly "unseen" from a data preparation perspective. While this also means that the preprocessing might not be fully optimised for the specific cross-validation training sets, this should not hurt the generalisability of the model which may even increase due to the use of a much bigger sample (342 593 discarded records). Abbreviations were seen as stylistic attributes of authors and less likely to be affected by the quality considerations discussed in Section 6.2.1 making the excluded records a viable abbreviation sample.

After running the abbreviation standardisation search-and-replace function on the final dataset, all remaining non-word characters were replaced with a single whitespace and the remaining text converted to lower case. In a final step, all excess whitespace was removed. These final steps could just as easily have been implemented in the tokenisation process inside the cross-validation. Although this would have no effect on the results, they were done before to avoid unnecessarily repeating the same procedure in every loop.

## 6.3 Data Description and Data Mining Objective

The final dataset selected for training contains 30 751 documents and five failure mode classes presenting relatively severe class imbalance as can be seen in Table 6.1 below. The F1-score is the most relevant evaluation metric as it accounts for both the precision and recall which is especially important for class imbalance. While no metric can adequately summarise the model performance, it is useful to have a single metric with which to compare and select models especially for optimisation. Both the macro- and micro-averaged scores are used in literature

**Table 6.1:** Class distribution in selected sample

| Failure Mode | 2007 | 2008 | 2009 | 2010 | 2011 | Total | Normalised |
|---|---|---|---|---|---|---|---|
| Electrical Failure | 169 | 390 | 329 | 745 | 246 | 1 879 | 6.11% |
| Electronics Failure | 664 | 1 696 | 921 | 1 819 | 1 944 | 7 044 | 22.91% |
| Hydraulic Failure | 325 | 875 | 972 | 485 | 370 | 3 027 | 9.84% |
| Mechanical Failure | 812 | 3 308 | 3 305 | 1 002 | 455 | 8 882 | 28.88% |
| Structural Failure | 117 | 66 | 180 | 4 592 | 4 964 | 9 919 | 32.26% |
| Total (Test Folds) | 2 087 | 6 335 | 5 707 | 8 643 | 7 979 | 30 751 | 100.00% |
| Training folds | 28 664 | 24 416 | 25 044 | 22 108 | 22 772 | | |

and present different views of the performance. It is desirable to maximise both, but for this implementation the micro-average was deemed more important as it is indicative of the performance of the largest number of documents. As per the project objectives defined in Chapter 1, validity is critical. Therefore the data mining objective is to maximise the micro-averaged F1-score and to demonstrate the validity of the results.

The final dataset still contains chronological data-drift as can be seen in Figure 6.2, which must be taken into account for evaluation measures that use the IID assumption.

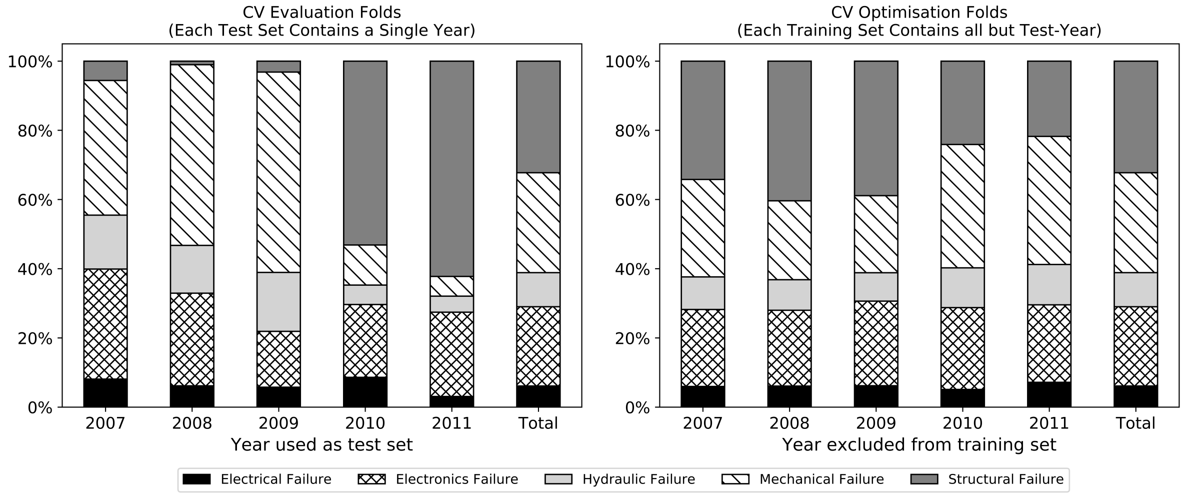


**Figure 6.2:** Final sample training and testing folds distribution. The test folds are indicative of the total sample

## 6.4 Experimental Design

This section presents the experimental design used to achieve the data-mining objectives identified in the previous section. As per the CRISP-DM methodology, the experimental procedure presented here is the outcome of several iterations used to review and refine the process. For the sake of brevity, only the final experimental procedure is presented here.

The design was guided by the literature reviewed in previous chapters including the common-practices identified from the related studies as well as the more theoretical concerns identified in Section 3.8.3. It includes Data Preparation, Modelling and Evaluation sections as per the CRISP-DM methodology, but to ensure the validity of the results, these three steps are embedded into a single, nested procedure as described in the Evaluation section.

It is important to note that the experiment was designed from a model assessment, rather than model building, perspective to ensure a conservative performance estimate. A better model can (and must) be learned from the total dataset before deployment so that the results from this analysis provide a lower limit on the generalisation performance that can be expected.

### 6.4.1 Optimised Preprocessing

This section describes the preprocessing steps that were optimised within the cross-validation process. As previously discussed, hyperparameters include any variable in the modelling process that is not estimated directly from the data (Tsamardinos *et al.*, 2015). This includes all user decisions including the selection of preprocessing transforms and parameters. To prevent data contamination resulting from optimising these data-preparation steps on the test data, the preprocessing phase is refined within the inner loop of a nested cross-validation (along with estimator hyperparameters) according to the optimization strategy described below.

This section is organized according to the main preprocessing functions, namely Tokenisation, Feature Weighting, Data Scaling and Feature Selection. It provides an overview of the range of preprocessing transforms evaluated for each stage as well as details of their implementation.

#### 6.4.1.1 Tokenisation

Tokenisation was performed using the Scikit-learn CountVectorizer module which performs both tokenization and vectorisation to yield a Document Term Matrix (DTM) of numeric feature vectors. Additional preprocessing transforms were evaluate using NLTK stop-words and Porter Stemmer (2).

A standard tokenization process was used to split the text into word-tokens treating whitespace as delimiters[2]. From the studies evaluated, no consensus could be found on the effectiveness of either stemming or stop word removal. Accordingly, the effect of including neither, one or both of stop-word removal and stemming was investigated.

For the stop word removal, the NLTK English stop-word list containing 153 English words was used. For stemming, the NLTK extension of Porter Stemmer (2) was used as per the recommendations by Gentzkow *et al.* (2017) who cites the Porter Stemmer as the industry standard for English. Only words containing more than two characters were passed to the stemmer leaving the rest unchanged (no sensible stem for non-standard abbreviations or numeric terms). When both stemming and stop-word removal are applied, the stop-words are removed first and the remaining text sent to the stemmer. This ordering is supported by Baharudin *et al.* (2010) and makes intuitive sense as running stop words through the stemmer only to be discarded directly thereafter is a computational waste.

To preserve some level of word-order information while retaining the simplicity and computational efficiency of the bag-of-words model, the inclusion of higher order n-grams were considered in addition to the standard unigram. This was limited to bigrams and trigrams because while researchers may disagree on the effect of including these, it is widely accepted that n>3 is not useful and may even decrease

[2] This is a typical tokenisation implementation but presents a slight modification to the default used by CountVectorizer which recognises only words with two or more characters. The default tokenizer would discard any remaining 1-character abbreviations as well as the numeric terms standardised to 0 in Section 6.2.2.

performance (Tan *et al.*, 2002). As per the recommendations by Bekkerman and Allan (2004), these were only considered as extension to the standard unigram (i.e. combined with) and not as replacement.

The vectorizer returns a Document Term Matrix constructed according to the vector-space-model where every column represents a feature and every row represents a document (transformed to a feature vector) so that every element represents the weight (importance) of the respective feature in the respective document.

#### 6.4.1.2 Feature Weighting

The weighting schemes evaluated were: binary indicators (features weighted equally indicating only the presence or absence of a term in a document), term frequencies (features weighted according to the frequency in a document) and sublinear log frequencies (features weighted according to the logarithm of the frequency in a document).

Term frequency (TF) is the predominant method found in literature. It has been demonstrated to be a very effective approach, especially when combined with IDF scaling (discussed below). However, most research has been focussed on documents longer than 100 words such as the Reuters-21578 dataset which has an average document length of 160 words in comparison to 54 in this dataset (both before stop word removal). Words are much less likely to repeat in such short documents leading to what Timonen (2012) calls the TF=1 challenge. He further states that weighting schemes based on TF do not perform well on short documents.

Therefore, less common variants binary and logTF were also evaluated. While Ikonomakis *et al.* (2005) recommend using binary features for computational reasons (citing equivalence with TF for short documents), the Scikit learn documentation recommends binary features as superior for short documents citing noisy TFIDF features which can lead to model instability[3]. Despite these recommendations, term frequencies are still the predominant weighting scheme used in short

[3] Furthermore, one of the algorithms evaluated (Bernoulli Naïve Bayes) requires Boolean features.

text classification. Hence, log frequencies were also evaluated as a middle-ground between the two as it dampens the influence of frequency somewhat.

#### 6.4.1.3 Data Scaling

To account for the effect of different document lengths, feature lengths and feature distributions additional scaling functions were evaluated for all the base values, namely Document Length Normalisation, IDF and feature standardisation. These were implemented using the Scikit-learn Normalizer, TfidfTransformer and StandardScaler modules.

From the data understanding phase, it was seen that the document length varied from 2-93 words per document. To prevent features in long documents from dominating the classification process, document length normalisation was investigated which scales each document (row in DTM) independently by the length of its feature vector. The feature length was computed using both L1 and L2 normalisation.

To account for the frequency component resulting from corpus frequency rather than document relevance, Inverse Document Frequency normalisation was investigated which scales each feature (column in DTM) by its IDF to yield IDF, TFIDF and logTFIDF for the respective frequency transforms.

For the support vector machines an additional feature scaling method was investigated that standardizes features to have a unit variance. This prevents features with greater numeric ranges dominating those with smaller ranges (a common problem for distance-based estimators such as SVM) which can improve both the performance and the convergence rate of the algorithm. It should be noted that this is different from the traditional standardisation which also centres the data around a zero mean as this would destroy the sparsity of the data (without which the dimensionality would be intractable) (Stolcke *et al.*, 2008). Furthermore, due to the natural sparseness of text, Aggarwal (2018) suggests that the attribute means are naturally close to zero meaning the lack of centring should not have a big effect (supported by Stolcke *et al.* (2008)). Although none of the studies evaluated in

Chapter 4 implement this approach, it is common in non-text domains where IDF is meaningless and was considered worthy of review.

#### 6.4.1.4 Feature Selection

Both supervised and unsupervised methods were used to evaluate the effect of selecting different sized feature subsets. For supervised feature selection, Chi-squared was implemented to rank features according to class dependence (correlation to the various classes) and select between 50-100% of the highest scoring features. For unsupervised feature selection, both high and low document frequency (DF) thresholding was evaluated to discard the most common and most rare terms respectively. For the lower threshold an absolute DF of 1-5 was evaluated (discarding features appearing in less than the threshold documents); and for the upper threshold a proportional DF of 95-100% was evaluated (discarding features appearing in more than the threshold proportion of documents).

Note that both the DF scores and Chi-squared statistics were only used to rank the features for subsequent elimination or inclusion, not to weight them. The feature values passed to the estimator is unchanged from the feature weighting steps above. The effects of feature selection were evaluated for all estimators as Forman (2004) demonstrated that contrary to popular belief, SVM can benefit from feature selection.

The focus of thresholding is feature elimination rather than selection as it attempts to discard the worst features (as opposed to selecting the best). For this reason, it was applied before data scaling so that the document length normalisation was not affected by the discarded features (likely to be outliers). In contrast, the Chi-squared statistic (which attempts to select the most informative features rather than discarding the worst) was performed as a final step before estimator-training so that the ranking can benefit from appropriately weighted (and normalized) data. The thresholding was implemented inside CountVectorizer and the Chi-squared using the Scikit-learn SelectPercentile module.

## 6.4.2 Modelling

As per the no-free-lunch theorem, there is no objectively "best" estimator for all problems and no way to predict with certainty which will be superior for a specific data set. Therefore, practitioners are typically guided by the results from related research as well as the specific properties of the various estimators which may make them more or less desirable for a certain objective. Accordingly, two broad estimator classes were selected for the study, namely Naïve Bayes and Support Vector Machines. For each of these, two variants were evaluated called multiNB, bernNB and SVC, linSVC respectively.

### 6.4.2.1 Naïve Bayes

Naïve Bayes is one of the most popular estimators due to its speed, efficiency and simplicity. Two variants of Naïve Bayes were evaluated, namely the Bernoulli Naïve Bayes (hereafter called bernNB) and the Multinomial Naïve Bayes (hereafter called multiNB). They differ only in the assumption they make about the distribution of the data with the Bernoulli assuming a multi-variate Bernoulli distribution and the Multinomial Naïve Bayes assuming a multinomial distribution.

Although the Multinomial is generally considered superior to Bernoulli, some studies have shown comparable and even superior performance by Bernoulli in certain situations – specifically for shorter documents (Wang and Manning, 2012). This is also supported by the Scikit-learn documentation.

These estimators require features according to the model assumptions. The Bernoulli Naïve Bayes require boolean features, hence only binary frequencies and no scaling was evaluated. The multinomial model was initially developed for word frequencies, however, several authors have demonstrated better performance using binary features while others using logarithmic frequencies. Furthermore, while the independence assumption means they are insensitive to feature scale, many authors have demonstrated that the excessive word burstiness can sometimes cause over sensitivity to word frequencies which can be improved with IDF weighting. However feature standardisation is not generally applied and not evaluated. Finally, several authors recommend document length normalisation for multiNB so this was also considered.

Both estimators were implemented using the Sciket-learn Naïve Bayes class which is a fairly standard Naïve Bayes implementation. However, these were set to assume uniform class priors rather than learn them from the data unlike standard implementations (in Sciket learn or otherwise). This was done because from the data understanding phase it was clear that models would be expected to handle significant data drift meaning class priors observed in the data was subject to change which could hurt, rather than benefit, performance. Uniform priors would assume equal probability for all classes which would prove more true in the long run and prevent over focussing on the majority class. This was hard-coded into the algorithm and not as parameter setting to be decided by hyperparameter optimisation. The reason this was hard-coded into the algorithm and not evaluated in the hyperparameter optimisation is because the optimisation CV was performed using stratified sampling which created artificially IID data to promote learning. However, this parameter is directly influenced by data distribution and optimisation would have optimised for IID data in which case learning priors is a good idea.

The only estimator parameter optimised in the hyperparameter optimization CV is the smoothing parameter alpha which accounts for words not observed during training. For both estimators a continuous, uniform distribution was created from which alpha values could be sampled in the randomized optimisation procedure described below.

#### 6.4.2.2 Support Vector Machines

Support Vector Machines are particularly well suited to the characteristically high-dimensional nature of text data as it is able to utilise the natural sparsity of text to avoid a dimensionality crisis with some even calling it best in class for text classification (Allahyari *et al.*, 2017).

While higher-order kernels have been used to great effect in other domains, they have not been shown to provide any discernible performance benefits for text classification leading only to an undesirable increase in model complexity (Lewis *et al.* 2004; Leopold and Kindermann 2002). Hence only linear Support Vector Machines were evaluated using the Scikit-learn SVM class.

In the remainder of the document, SVM and SVC is used interchangeably in keeping with the Scikit-learn notation of abbreviating Support Vector Classification (as opposed to Support Vector Regression) to SVC. This is mostly used when referring to the specific estimators implemented rather than the general estimator class.

Two variants of the linear SVM were implemented that use different approaches to solve the optimisation problem. Although they solve the same problem and should theoretically be equivalent, the heuristic nature of the approaches mean that can sometimes lead to quite different results. The first approach, denoted SVC was implemented using Scikit-learn SVC class that uses the LIBSVC solver. This is a general SVM library that can handle higher order-kernels and uses a Sequential Minimal Optimization (SMO) algorithm.

The second approach, denoted linSVC, was implemented using the Scikit-learn linearSVC class that uses the LIBLINEAR solver that implements a coordinate descent algorithm. It is a specialised linear solver which does not offer any higher order kernels and has therefore been optimised for the linear case making it more efficient for large datasets. The theoretical basis for the different implementations is beyond the scope of this report, but the interested reader can refer to the respective documentation for more details.

Neither approach is inherently multiclass but follows different multiclass decomposition strategies by default. The liblinear uses a one-vs-rest while the SVC uses a one-vs-one. Several studies have been done to compare these approaches with no definite best answer. The biggest difference is in the computational time as the one approach trains k models using all the data, while the other trains more models but uses a smaller number of instances in each. The co-ordinate descent method is insensitive to the number of samples with computational complexity stemming only from the feature space making OVR preferable. In contrast, the SMO is sensitive to the number of samples making the OVO which trains more models but on a smaller training sample for each, preferable.

For both estimators, two algorithm parameters were optimised in the hyperparameter optimisation, the regularization parameter C and the class weight scheme. The regularization parameter impacts the trade-off between model complexity and incorrect classification. The model complexity depends on the number of documents used to form support vectors and greatly affects the training time. Hence low values of C is preferred. Accordingly, a log uniform distribution was created between $10^{-5}$ and $10^{2}$ using SCIPY distributions. The continuous distribution allows for finer tuning than grid search and the log uniform distribution is used to skew the sampling to lower C values while still allowing for larger values to be tested.

The final parameter optimised in the hyperparameter optimisation is the class weights where both balanced and none were evaluated. The balanced distribution adjusts the class weights inversely proportional to its frequencies to artificially balance classes preventing SVM from learning only the majority class as SVMs are sensitive to class imbalance.

This is different to the class priors in Naïve Bayes. While related they make different assumptions. Naïve Bayes is a probabilistic, generative model with the priors making an assumption about the class distribution that can be expected in future. SVM is a distance-based, discriminant model which uses class weights to make an assumption about the importance of minority classes vs majority classes.

### 6.4.2.3 Hyperparameter Optimization

The hyperparameter optimization was performed using a randomized search procedure over the parameter space specified for each estimator as recommended by Bergstra and Bengio (2012) in Section 3.7.3. As previously discussed, the advantage of this method over the more common exhaustive grid-search is two-fold. Firstly a much larger range of parameters settings can be explored for a fraction of the computational effort. Secondly, the range of values to evaluate for each parameter can be provided as a continuous distribution which enables a much finer optimisation than is possible for grid-search which can only evaluate a discrete set of values (Bergstra and Bengio, 2012).

The optimisation was implemented using the Scikit-learn RandomizedSearchCV module with the random state set to 42 for repeatability. In this implementation, a set number of parameter combinations are randomly sampled (with replacement) from the specified parameter-space and evaluated using a cross-validation loop. Each fold is used as a testing-set once while the other folds are used to train a model for each of the parameter settings. The parameter-settings of the model achieving the highest mean performance over all folds is used to train a final, optimised model on the full dataset (using all folds). As per the recommendations by Zheng (2015), the optimisation was set to evaluate 60 parameter-combinations.

From the data understanding phase, it became apparent that the data has significant class imbalance. To ensure that every fold has sufficient training and testing data from each class, stratified sampling was used for the optimisation CV as per the recommendations by Tsamardinos *et al.* (2015), Santafe *et al.* (2015) and others. The Scikit-learn StratifiedKFold module was used to implement the RandomizedSearchCV cross-validation and set to include shuffling with the random state set to 42 for repeatability.

As per the evaluation design discussed below, to prevent an optimistically biased performance estimate, the hyperparameter optimization was performed in the inner loop of a 5x2 nested cross-validation. In other words, the total optimisation process was performed 5 times, once for each outer evaluation fold with the remaining folds passed as training data to the inner, 2-fold optimisation process so that the inner optimisation loop only ever received approximately 80% (four folds) of the data.

However, it is important to note that this design was selected from a model assessment, rather than model building, perspective to ensure that the model performance is over, rather than under estimated. A better model will be learned by performing a single, 5-fold optimisation to find the best parameter-settings with which to train a final, optimised model on the full dataset. This is the model that will typically be deployed in practice with the nested-CV results expected to be a conservative estimate of the true generalisation performance.

### 6.4.3 Evaluation

This section describes the evaluation process used to evaluate the model performance. As per Section 3.7.4.2, whilst important, the predictive performance is not the only measure of model success. Other important factors include interpretability, actionability and efficiency. Interpretability was addressed by the consideration of the white-box Naïve Bayes algorithm and the exclusion of dimensionality reduction methods. Actionability was addressed through communication with the data provider to ensure that the data mining objectives aligned with their business objectives. In terms of efficiency, while always desirable, for this application neither training nor prediction speed is of utmost importance. Computational complexity and memory footprint is also not very constrained so that the efficiency is limited only by practicality.

In terms of the predictive performance, the micro-averaged F1-score was identified as the most important evaluation metric. However the macro-averaged and per-class F1-scores as well as a confusion matrix and training time was also considered.

It is generally agreed that in the absence of sufficient data to sample multiple independent training and testing sets, CV is a preferable evaluation procedure to resubstitution, two-way train/test split or even a three-way train/validation/test split. Although most of the domain specific studies evaluated in Section 4.1 do not make use of CV to evaluate different estimators, the vast majority of broader empirical studies do. However, on the specific implementation of CV, as well as on the aggregation of results obtained from the separate folds, there is less consensus. The main discrepancies identified from research is:

- Sampling strategy: Random, Stratified or Grouped (Blocked)
- Nested or single-loop cross-validation
- Obtaining final performance estimates by averaging the metrics achieved for each fold or by determining metrics for aggregated predictions
- Statistical quantification of results

Many authors do not explicitly address these issues and provide only the selected methods with no explanation or motivation of their choice, while others do not even provide sufficient details of their process to enable replication. As per Forman and Scholz (2010) this is not necessarily with intent, but rather due to a lack of awareness of different options and possible different interpretations. The papers that did address these issues, are mostly the theoretical studies with a very specific focus and none addressed all of these concerns.

Hence, the final evaluation procedure was designed according to the most conservative theoretical papers. Because no empirical studies were found in the domain (or elsewhere) applying this exact procedure, the estimators were also evaluated using more common methods to demonstrate the difference between these methods and to evaluate the validity of the assumptions inherent in every evaluation method. These results are discussed in Chapter 8.

#### 6.4.3.1 Evaluation Procedure

To prevent data-contamination resulting from over-optimising the estimators for the specific dataset, the estimators were evaluated using 5x2 nested cross-validation with an inner optimisation loop and a outer evaluation loop as per the recommendations by Tsamardinos *et al.* (2015) and Krstajic *et al.* (2014).

From the data understanding and exploration phase it became apparent that the IID assumption does not hold for the dataset under consideration. Although the data preparation phase includes the selection of a more homogeneous sample, there is still clear distributional changes with time and possibly other unknown variables as well.

To account for the data-drift present in the available dataset (and which the models will accordingly be subjected to in practice) the models were evaluated using blocked cross-validation rather than stratified cross-validation as per Bergmeir and Benitez (2012) in Section 3.8.3. Rather than separating the folds with artificially even class distributions (stratified CV), the data was separated into chronological folds grouped by year. In other words, each year constitutes one fold so that all but one year's data is available for training in each evaluation. The training sample selected in the data preparation phase spans over five years with clear distributional differences between the beginning and the end. While the distributional variation between years is significant, each year contains a reasonable amount of samples from each class. Hence, the years form a convenient blocked splitting strategy to be used in Bergmeir and Benitez's (2012) blocked cross-validation and also corresponds to the final-year hold-out test-set of Lewis *et al.* (2004).

Because the dataset presents significant class imbalance, the two-fold inner optimisation loop was split using stratified sampling to ensure all training and testing samples contain sufficient minority class samples as per the recommendations by Tsamardinos *et al.* (2015), Santafe *et al.* (2015), Forman and Scholz (2010) and others. This is especially important for the smaller sample sizes available to the inner cross-validation loop. For the optimisation process, only the relative performance is of interest and while the IID assumption violation may hurt the ability to optimise for changing distributions, it does not affect the validity of the performance evaluation. Here stratified sampling was deemed more appropriate to ensure sufficient samples from each class available for training.

In addition to the algorithm hyperparameters, the optimisation step also includes the preprocessing design decisions such as feature weighting, scaling and selection. The optimisation was performed using a randomised search procedure that performs 60 iterations of the 2-fold CV over the parameter search-space provided to find the parameter-set that maximises the geometric mean of the micro- and macro-averaged F1-score (see metrics discussion below). The optimal parameter-set is used to re-train a model on the full optimisation data set (outer loop training data) and is evaluated on the unseen outer-loop test data.

This means that for each of the four algorithms (multiNB, bernNB, linSVC and SVC) a total of 605 models were trained[4] – 600 for the parameter optimisation and a further 5 for the evaluation.

[4] For each outer training fold 120 models (2x60) are trained in the optimisation CV phase with the resulting optimised model retrained on the full training set for evaluation in the outer fold. This process is repeated 5 times in the outer evaluation CV namely 5x(120+1) = 605 models.

### Metrics

As previously discussed, the micro-averaged F1-score was deemed the most appropriate evaluation metric. The class-imbalance makes Accuracy an inappropriate metric and, because it is a multiclass problem, other popular classification metrics such as ROC and AUC are also not applicable. Therefore the models were evaluated using the harmonic mean of the precision and recall, namely the F1-score as defined in Section 3.7.4.1.

The micro-averaged F1-score is more representative of the true model performance than the macro-averaged score as the micro metric is evaluated per-document (each document is weighted equally) and accordingly the majority class performance tends to dominate the score. This is desirable as the majority class is the most frequently occurring class by definition. Hence, a model performing poorly on the majority class, will perform poorly on the majority of documents. Of course, good performance on the smaller classes is also desirable and to prevent the micro-averaged score from hiding unacceptably poor performance in the smaller classes, the macro-averaged F1-score, per-class F1-score and confusion matrix of each classifier were considered as well.

For the automated optimisation of hyperparameters, a single score value is needed to guide the optimisation process. Although the micro-averaged F1-score was selected as the most pertinent evaluation metric, this should not be at the cost of the complete disregard for the other metrics. To ensure a more balanced estimator, the optimisation was performed to maximise the average of the micro- and macro-averaged F-scores. The macro-average is evaluated per-class (each class is given the same weight) and leads to the scores of the minority classes being up-weighted beyond their proportional representation in the total data set. By taking the mean of the micro- and macro- averaged F-scores, the majority class dominance is maintained by the micro-averaged score (where the majority class is up-weighted according to its proportionality), but lessened by the inclusion of the macro-average (where the majority class is weighted equally to the minority classes). Although no papers were found employing this method, as Dietterich (1998) states, the inner loop of the cross-validation function is only concerned with relative performance in order to guide parameter selection. Therefore questionable methodology may hamper the performance but does not affect the validity of the results.

Initially the arithmetic mean of the micro- and macro-average was taken as the optimisation metric so that: $F_{opt} = 0.5(F_{min} + F_{mac})$. However, this produced very unstable results as each optimisation fold selected different parameter sets. High variability in the parameter selection indicates instability of the modelling process which leads to high variance in the evaluation results which is undesirable regardless of the final performance achieved. Upon closer inspection, it became apparent that the arithmetic mean was frequently dominated by extreme values, namely high micro scores corresponding to low macro scores. Therefore, the optimisation metric was changed to the geometric mean of the micro- and macro-averaged scores (calculated from $\sqrt{F_{mic} * F_{mac}}$) which is less sensitive to high outliers than the arithmetic mean (while also less sensitive to low outliers than harmonic mean) as can be seen in Figure 8.6.

The use of statistical tests to evaluate and compare machine learning models is somewhat controversial and no consensus could be found from literature regarding how, or even whether, to perform such tests (Section 3.8.2). Since the focus of this experiment was feasibility, and not to prove whether a certain technique is better than another, it was decided that simply choosing the method with the best cross-validated performance is sufficient as per the recommendations by Witten *et al.* (2011). The final cross-validated F1-score was computed according to recommendation by Forman and Scholz (2010) to aggregate the predictions from all folds before calculating a single F1-score (called $F_{AGG}$), rather than averaging the F1-scores achieved by each fold (called $F_{AVG}$) (Section 3.8.3).

To get an indication of the model stability and range of performances that can be expected in practice, the models were also evaluated according to their standard deviations. However, since the aggregated F-score provides only a single point estimate, the standard deviations were estimated from the per-fold F1-scores. It is therefore important to keep in mind that these per-fold F1-scores, and accordingly the standard deviations calculated from them, may contain significant levels of bias due to the ill-defined edge-cases which may manifest as exaggerated variation.

# Chapter 7

# Empirical Analysis: Results

The experimental procedure described in the previous chapter was implemented on the selected data sample described in Section 6.2.1. The most important results are provided in this chapter including the evaluation of both the models and the modelling process used to obtain them. This forms part of the Evaluation phase of the CRISP-DM methodology.

It starts with the results from the hyperparameter optimisation to investigate the model (and modelling process) stability and to compare the parameter selections with those found in literature. Next, the random baseline performance of this dataset is modelled to provide a frame of reference for the model performance evaluated in the section below. To gain a better understanding of the respective model behaviours, their learning curves are then evaluated to diagnose the main source of error and find the most promising areas of improvement for each model. As discussed in Section 3.8.2, no statistical tests were performed in this evaluation and the word significant is only ever used in its generic sense and not intended to convey any statistical meaning.

## 7.1 Hyperparameter Optimisation

The purpose of the hyperparameter optimisation was to improve model performance, not to investigate the details of specific parameter behaviours and interactions. Therefore, these results were only inspected in a cursory manner to assess the model stability and ensure sensible outputs. The most important observations are provided here along with a speculative discussion for possible explanations of the results, but a detailed investigation of the various parameter interactions is left for future work.

The results are summarised in the tables below with Table 7.1 providing the optimised preprocessing parameters and Table 7.2 the optimised algorithm parameters. The inner columns provide the parameters selected for each outer fold of the nested cross-validation and are labelled with their corresponding test year (which was left out of the sample sent to the inner optimisation loop). The second to last column summarises the per-fold results using a voting strategy to show the over-all best selection of parameters according to the nested-cross-validation (NCV). These can be considered the “average” parameter values used to obtain the average performance values in Section 7.3.

As mentioned before, the NCV experimental design is only for evaluation purposes. For actual deployment the parameters would be optimised on the whole dataset using a single loop CV and the optimal parameters then used to train a final model on the full dataset to maximise the amount of training data. According to Forman (2007) the optimality of parameters may be sample-size dependent leading to the evaluation of suboptimal models by the NCV which could lead to underestimated performance estimates. To evaluate the stability of the parameter optimisation for different sample sizes, it was repeated using a non-nested, single loop cross-validation with the same splits as before so that the parameters are optimised on the full training folds. These parameters are provided in the final column of the tables below (CV Opt.) and the methodological implications of these results discussed in Section 7.1.1.

All values not corresponding to average NCV selection (NCV Opt.) are indicated in red. Where relevant, the default values are shown underneath the parameter name in the first column. All empty cells are greyed to distinguish between parameters that were not considered at all and those excluded by the optimisation process (indicated with a dash).

These results provide information on both the model and modelling process stabil-

**Table 7.1:** Preprocessing hyperparameter optimisation results - Year columns are the (unincluded) test folds of the respective training folds used to optimise parameters

| Pre-Processing Parameters | | 2007 | 2008 | 2009 | 2010 | 2011 | | NCV Opt. | CV Opt. |
|---|---|---|---|---|---|---|---|---|---|
| N-gram range (1-1) | multiNB | (1 - 3) | (1 - 3) | (1 - 3) | (1 - 3) | (1 - 3) | | (1 - 3) | (1 - 3) |
| | bernNB | (1 - 3) | (1 - 3) | (1 - 3) | (1 - 3) | (1 - 3) | | (1 - 3) | (1 - 3) |
| | linSVC | (1 - 2) | (1 - 2) | (1 - 2) | (1 - 2) | (1 - 2) | | (1 - 2) | (1 - 2) |
| | SVC | (1 - 2) | (1 - 2) | (1 - 2) | (1 - 2) | (1 - 2) | | (1 - 2) | |
| Stop-Words | multiNB | - | - | - | - | - | | - | Stop |
| | bernNB | Stop | Stop | Stop | - | - | | Stop | Stop |
| | linSVC | - | - | - | - | - | | - | - |
| | SVC | - | - | - | - | - | | - | |
| Stemming | multiNB | Stem | - | Stem | - | Stem | | Stem | - |
| | bernNB | - | - | - | Stem | Stem | | - | - |
| | linSVC | Stem | Stem | Stem | Stem | Stem | | Stem | Stem |
| | SVC | - | - | - | - | - | | - | |
| Max DF (None) | multiNB | 96.6% | 95.5% | 96.6% | 95.2% | 96.6% | | 96.6% | 95.5% |
| | bernNB | 96.5% | 96.5% | 96.5% | ≈100% | ≈100% | | 96.5% | 96.5% |
| | linSVC | 96.6% | 98.2% | 96.6% | 96.6% | 96.6% | | 96.6% | 96.6% |
| | SVC | 96.3% | 96.3% | 96.3% | 95.9% | 95.9% | | 96.3% | |
| Min DF (None) | multiNB | 1 | 3 | 1 | 3 | 1 | | 1 | 4 |
| | bernNB | 2 | 2 | 2 | 1 | 1 | | 2 | 2 |
| | linSVC | 3 | 4 | 3 | 3 | 3 | | 3 | 3 |
| | SVC | 4 | 4 | 4 | 4 | 4 | | 4 | |
| Frequency (TF) | multiNB | TF | Bool | TF | Bool | TF | | TF | Bool |
| | bernNB | Bool | Bool | Bool | Bool | Bool | | Bool | Bool |
| | linSVC | TF | logTF | TF | TF | TF | | TF | TF |
| | SVC | LogTF | LogTF | LogTF | LogTF | LogTF | | LogTF | |
| Feature Normalisation | multiNB | - | - | - | - | - | | - | - |
| | bernNB | | | | | | | | |
| | linSVC | IDF | IDF | IDF | IDF | IDF | | IDF | IDF |
| | SVC | IDF | IDF | IDF | - | - | | IDF | |
| Doc-length Normalization(L2) | multiNB | - | L1 | - | L1 | - | | - | L2 |
| | bernNB | | | | | | | | |
| | linSVC | L1 | L1 | L1 | L1 | L1 | | L1 | L1 |
| | SVC | L2 | L2 | L2 | L2 | L2 | | L2 | |
| Feature Selection CHI2 | multiNB | 56.9% | 53.5% | 56.9% | 80.9% | 56.9% | | 56.9% | 94% |
| | bernNB | 68% | 68% | 68% | 58.3% | 58.3% | | 68% | 68% |
| | linSVC | 68.5% | 51.3% | 68.5% | 68.5% | 68.5% | | 68.5 | 68.5% |
| | SVC | 51.1% | 51.1% | 51.1% | 88.4% | 88.4% | | 51.2% | |

ity which are discussed in Section 7.1.1 below. The specific parameter-selections are briefly discussed after that and compared with those used in literature.

**Table 7.2:** Estimator hyperparameter optimisation results - Year columns are the (un-included) test folds of the respective training folds used to optimise parameters

| Algorithm Parameters | | 2007 | 2008 | 2009 | 2010 | 2011 | | NCV Opt. | CV Opt. |
|---|---|---|---|---|---|---|---|---|---|
| Alpha - Naïve Bayes(1) | multiNB | 0.7 | 0.2 | 0.7 | 0.1 | 0.7 | | 0.7 | 0.7 |
| | bernNB | 0.7 | 0.7 | 0.7 | 0.3 | 0.3 | | 0.7 | 0.7 |
| Penalty C - SVM(1) | linSVC | 0.23 | 0.2 | 0.23 | 0.23 | 0.23 | | 0.23 | 0.23 |
| | SVC | 0.39 | 0.39 | 0.39 | 1.14 | 1.14 | | 0.39 | |
| Class Weight - SVM (Uniform) | linSVC (OVR) | Bal. | Bal. | Bal. | Bal. | Bal. | | Bal. | Bal. |
| | SVC (OVO) | Uni. | Uni. | Uni. | Uni. | Uni. | | Uni. | |

### 7.1.1 Stability

If the parameter selection changes significantly between folds (per row) it is an indication of instability in either the model, the modelling process or both. From the tables it can be seen that LinSVC is the most stable selecting identical parameter sets for all folds except the one excluding 2008. (Note that the column labels are the test sets not used for optimisation, hence column 2008 represents parameters optimised on data from 2007, 2009, 2010 and 2011.) Both bernNB and SVC seem to have two distinct groups that present more cohesive parameter sets, namely folds excluding 2010/2011 and those including them. This corresponds to the changing class distribution identified in Section 6.2 whereby 2007-9 and 2010-11 form more distinctive groups.

At first glance, MultiNB also seems to present two groups arbitrarily divided according to the exclusion of 2007, 2009, 2011 and the exclusion of 2008, 2010. However, only the first group (2007, 2009, 2011) is actually cohesive. The real value parameters such as alpha and percentile-features differ for 2008 and 2010 indicating that these folds are not so much a distinct group responding to changes in the data, but rather two separate instances of model instability. It is interesting to note that this corresponds somewhat to LinSVC that only presents instability for the fold excluding 2008. This may indicate a deviation in data quality (up or down) for 2008 so that the exclusion of those records leads to different results although it is not clear why this would be.

The parameter stability of the different models support their respective variance diagnoses made from the learning curves in Section 7.4. LinSVC and multiNB which has the lowest and the highest parameter instability, also have the lowest and the highest variance respectively. Furthermore, it can be seen that multiNB is the only model whose parameters are affected by the increased training data available to the single loop CV. This confirms both the high variance and the growth potential of multiNB as discussed in Section 7.4. The mostly consistent parameter sets between NCV and CV indicate that while more data would be beneficial, the optimisation process is sufficiently stable for the amount of data available to the NCV optimisation.

While model stability is desirable, so is robustness to concept-drift. From the per-year distribution of the changing parameters it seems that while BernNB and SVC are more stable than multiNB; multiNB is more robust to concept-drift as its parameter changes are not grouped chronologically and do not correspond to the distributional changes observed in Section 6.2. LinSVC is both more stable and more drift-robust than any of BernNB, SVC or multiNB; probably because of its low variance as discussed in Section 7.4.

### 7.1.2 Parameter Discussion

None of the models performed best with single word tokens in contradiction with Tan *et al.* (2002) and Bermingham and Smeaton (2010) who found that higher-order n-grams do not work well on short documents. This might be due to the highly specific, technical nature of the maintenance documents leading to a more limited corpus vocabulary than is normal (Section 6.2) which, according to Bekkerman and Allan (2004), increases the potential value of higher order n-grams as they are more likely to repeat than for large vocabularies.

What was interesting is that, except for bernNB, none of the models preferred stop-word removal which is very commonly recommended. This makes sense considering the low density and non-typical distribution of stop-words observed in the data. The low density of stop-words mean the purported benefit of stop-word removal is less, and the unusual distribution, specifically pertaining to the high frequency of *not* in comparison to standard English, indicates greater than usual significance of the word *not* (for instance in records describing *not dispensing* vs *dispensing fine*) so that stop-word removal hurts the performance. It may be worthwhile to remove *not* from the stop-word list and reevaluate the parameter selection in future work.

Stemming, which is more frequently advised against in literature, proved more successful, especially in linSVC and more moderately in multiNB and bernNB. Interestingly enough, bernNB only performed stemming for folds without stop-word removal meaning the popular combination of stop-word-removal and stemming was not selected by a single model for a single fold. This might be as a result of the short document length as Bermingham and Smeaton (2010) and Kobayashi *et al.* (2018) state that stop-word removal and stemming are not as effective for short documents.

The maximum document frequency threshold is surprisingly consistent for all models and within the range typically found in literature. The minimum DF threshold was relatively surprising as it was expected that the higher order n-grams selected by the Naive Bayes models would correspond to higher thresholds to handle the increasingly long-tailed distribution and large vocabulary size. However, both SVM implementations (which selected only bigrams) used higher thresholds than the Naive Bayes estimators (which selected trigrams). In hindsight, this can be explained by the fact that trigrams are much less likely to repeat than uni- or bigrams meaning lower thresholds are needed to benefit from their inclusion (a trigram occurring in two documents is likely more meaningful than a bigram occurring in two documents).

Both multiNB and linSVC selected term frequency features over either Boolean or logTF. This is in contrast with Rennie *et al.* (2003), Wilbur and Kim (2009), Bermingham and Smeaton (2010), Timonen (2012), Ikonomakis *et al.* (2005) and many others who argue against the use of TF features for short documents or otherwise. However, the mixed selection of feature representations is consistent with Wang and Manning (2012) who found that the difference between the various representations is much less significant for short documents.

Even more surprising was the bad performance of IDF for multiNB, in direct contrast to the recommendations by Wilbur and Kim (2009) to discard TF but use IDF. Only linSVC demonstrated clear benefit from IDF for all folds with SVC preferring it in three out of five. LinSVC favoured L1 norm while only SVC used

L2 norm. As expected, both implementations of SVM responded better to scaled data (both document length and feature wise). Another interesting result is that only the folds using Boolean features selected document length normalisation for multiNB while the majority of folds which use TF (and would have been expected to be more sensitive to document length differences) perform no normalisation in contrast to the results by Rennie *et al.* (2003). This might be because of the limited information in short documents. Without normalisation, features in longer documents dominate those in shorter documents. While this is usually undesirable, it may be that the slightly longer documents are more informative so that favouring their features is useful. However looking at the single-loop CV optimisation results, the multiNB parameters are much less surprising selecting Boolean features, stop-word removal and L2 normalisation which are more typical parameters. This is an indication that the unexpectedness of the NCV parameters for multiNB are as a result of too little data (suboptimal parameters) rather than the consequence of the specific data properties.

All the models implement substantial feature selection, even SVM which many have reported are robust to feature selection. Although varying by fold, multiNB and bernNB select the same smoothing parameter over-all. All of the selected smoothing parameters are below the default Laplace smoothing of 1 which is supported by Wilbur and Kim (2009) (and in contrast with the large number of authors who do not consider optimising Naive Bayes). The penalty parameters selected by the SVM are also both below the default but still in a similar range. As could be expected, linSVC consistently preferred balanced class weights while SVC preferred uniform. Because the linSVC implements an OVR scheme it deals with much more drastic class imbalance while the SVC implements OVO resulting in much more balanced datasets.

## 7.2 Baseline Performance

This section evaluates the maximum baseline performance that can be attributed to random chance for the dataset under consideration. This is used as a point of reference with which to compare the performance of the true models in all subsequent sections. The models are evaluated and compared with each other according to their respective F1-scores, and specifically, their micro-averaged F1-score which is identified as the most important metric in Section 6.4.3. Accordingly, this is also the focus of the baseline analysis only reporting the macro-averaged and per-class F1-scores to provide a more in-depth perspective of the different model behaviours.

The data presents relatively severe class imbalance with the most extreme ratio between the largest and smallest class being 84%-16% if only those two classes are considered. However, because it is a multi-class problem, the dominance of the majority class (*Structural Failure*) is softened to an overall 32% by the inclusion of several in-between class sizes. For this reason, both a majority (single-class) and a stratified dummy estimator were considered. Furthermore, upon the evaluation of the real models in Section 7.3, it was observed that all of them performed best on the same class: *Electronics Failure*, which is not the majority class. Therefore, an additional single-class dummy estimator was trained that always predicts *Electronics Failure* to get the maximum random performance that can be achieved for that class.

These models were evaluated with the re-substitution error computed from the full, shuffled dataset. In other words, contrary to the experimental design described in Chapter 6, not only was the same data used for training and testing (no unseen test data), but it was also shuffled before to create an artificially IID distribution. While this is not an appropriate evaluation strategy for real models, it is suitable for calculating the performance-baseline for two reasons.

Firstly, unlike the real models, dummy estimators are not intended for implementation. They provide a minimum performance threshold which real models must surpass to have value. This means that while a conservative evaluation of real models will underestimate rather than overestimate the performance, the opposite is true for baseline models. This makes the training error (which is prone to overestimating the performance) a conservative estimate of the performance that can be attributed to chance.

Secondly, because dummy estimators make predictions on a purely statistical basis ignoring both the training and the testing inputs, there is no real distinction between seen and unseen data. This means that unlike real models, they cannot overfit to the sample-specific relationships between document features and class labels found in the training data making the difference between the training and testing performance much less significant. Furthermore, as discussed in Section 6.2, the IID assumption does not hold for this data. This has an undesirable effect on the statistical formulation of both the stratified and majority class estimator when evaluated with the unshuffled, grouped CV that is centred around maintaining this inconsistency. Because the class distributions vary quite drastically between the training and testing sets of each fold (Figure 6.2) it leads to very poor hold-out performance for the stratified and majority class estimator which simply mimics the training distributions when making predictions. This is especially detrimental to the majority class estimator as the dominant class changes from one fold to the next. This effect can be seen for the grouped CV in Figure 6.2 where not only do none of the folds have matching training and testing majority classes, but the majority class selected for each training fold is actually a minority class ($<$10%) in the corresponding test fold. This results in micro and macro averaged scores below 6%. The effect is similar, but less extreme, for the stratified approach (the *Electronics Failure* single-class estimator results are independent of the evaluation strategy).

While this presents a more realistic scenario of the random performance that can be expected in changing distributions, as mentioned before the purpose of a baseline is not to be realistic. Therefore the shuffled re-substitution error was used to find the maximum possible performance that can be attributed to chance with which to evaluate the value of the true models.

The stratified estimator randomly assigns labels to the test-set while maintaining the class distribution observed in the training-set. The accuracy of these predictions relies on the accidental alignment of the predicted and true labels of the test set. This means that even for identical distributions (as for re-substitution error) the stratified estimator can theoretically achieve 100%, 0% and anything in-between. Therefore, the stratified performance was obtained from multiple repetitions with different random seeds to obtain the average estimate.

This was not necessary for the single-class estimators (majority and *Electronics*) which provide completely stable results. Their performance depends only on the prevalence of the respective class in the total dataset. Because all documents are put into one class, it necessarily leads to perfect recall of that class but zero for all others. Likewise, the precision of that class will always be equal to its proportion in the full dataset but zero for all others. This is true regardless of the order of the data making shuffled repetitions unnecessary.

The micro, macro and per-class F1-scores of all three models are shown in Figure 7.1 below. The scores of the majority and Electronics single-class estimators are indicated with a single marker and dashed line as indicated in the legend. Sixteen iterations were performed for the stratified estimator and the mean results are indicated with the black diamonds and bar graph. The iteration scores are indicated with the faded markers showing the tight spread around the mean. These were evaluated with random states set to 0, 1, 2... 15 to ensure repeatability. The mean score values are provided in Table 7.3 with the highest value for each metric highlighted.

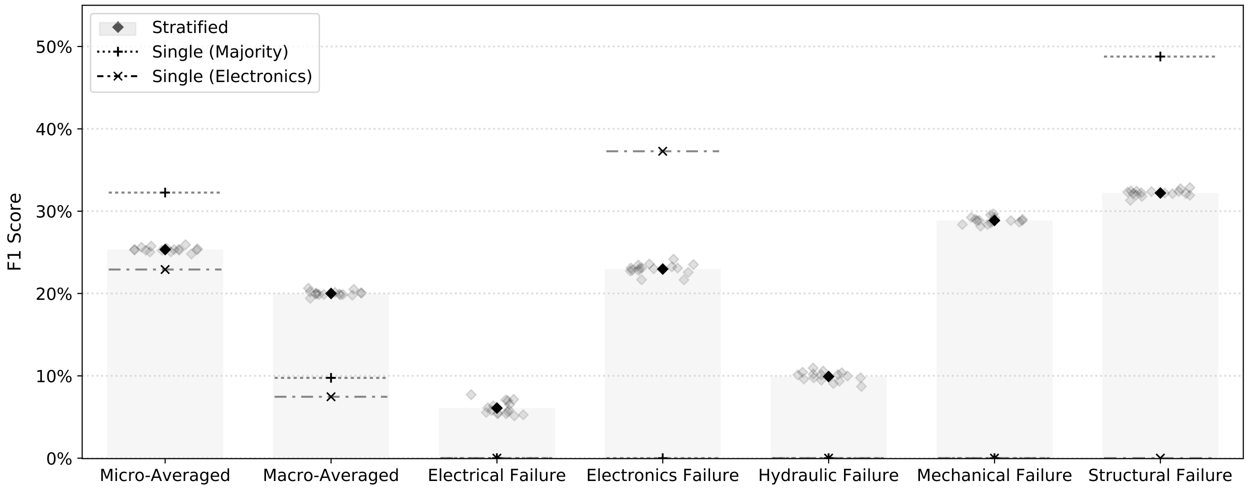


**Figure 7.1:** Micro, macro and per-class F1-scores of the baseline dummy estimators

The single-class estimators have the highest per-class scores for *Electronics* and *Structural Failure* respectively, but the stratified estimator is superior for all other

**Table 7.3:** Dummy estimator results, baseline performance (F1)

| Dummy Estimator | Micro Averaged | Macro Averaged | Electrical Failure | Electronics Failure | Hydraulic Failure | Mechanical Failure | Structural Failure |
|---|---|---|---|---|---|---|---|
| Stratified | 25,34<br>±0, 27 | 20,01<br>±0, 28 | 6,09<br>±0, 79 | 22,97<br>±0, 64 | 9,92<br>±0, 58 | 28,87<br>±0, 41 | 32,21<br>±0, 37 |
| Single-Class Majority (Structural Failure) | 32,26<br>±0 | 9,76<br>±0 | 0<br>±0 | 0<br>±0 | 0<br>±0 | 0<br>±0 | 48,78<br>±0 |
| Single-Class Electronics | 22,91<br>±0 | 7,45<br>±0 | 0<br>±0 | 37,27<br>±0 | 0<br>±0 | 0<br>±0 | 0<br>±0 |

classes for which the single-class estimators of course achieve zero. Because of this, the stratified estimator also has much higher macro-averaged result as this metric weights all classes equally and is therefore significantly downweighted by the predominance of zero-scoring classes for the single-class estimators. Since both *Structural* and *Electronics Failure* are fairly big classes they have reasonably high micro-averaged results as it weights all documents equally and is therefore dominated by the bigger classes. In fact, the majority-class estimator has the highest micro-averaged score, beating even the stratified estimator, as a result of its high performance on Structural Failure which is of course the largest class and accordingly has the most influence on this metric.

While the micro-averaged F-score was identified as the most important metric in Section 6.4.3, the stratified estimator (which has the second highest micro score) was selected as the baseline as it has better over-all performance and provides a non-zero point of reference for all metrics. These scores are indicated with a solid line in all ensuing results plots. For the metrics where one of the single-class estimators surpass the stratified results, the single-class scores are also provided as reference and are generally indicated with a dashed line. However, this provides another point of reference separate to the stratified baseline and should not be used without remembering their accompanying low scores. While it is certainly desirable for the true models to surpass the combined highest scores from all the dummy estimators, this would no longer be a random baseline.

Only the mean scores are used as frame of reference in the sections below as the variation around the means are inconsequential compared to that of the true models. The single-class estimators are stable by definition and the stratified scores are tightly distributed around the mean with a standard deviation below 1% for all metrics.

## 7.3 Model Performance

In this section, the models are evaluated and compared with each other according to their respective F1-scores and to a lesser extent their efficiency. The micro-averaged F1-score is identified as the most important metric and is accordingly the focus of the results (called the evaluation metric). The macro-averaged and per-class F1-scores as well as the model efficiency is further considered to provide a more in-depth perspective of the different model behaviours.

The models are evaluated in the outer loop of the nested cross-validation as described in Section 6.4.3. The most important results are summarised in Table 7.4 and plotted against the baseline performance in Figure 7.2. Each marker represents the mean F1-score achieved over all the folds calculated using the $F_{AGG}$ method described in Section 3.8.3. The error-bars indicate one standard deviation from the mean to give an indication of the stability of the model (and modelling process) as well as giving a better estimate of the range of expected performance. These are calculated from the per-fold scores.

The horizontal lines provide the baseline performance of each score (micro, macro and per-class) as achieved by the dummy estimators described in Section 7.2. The solid lines represent the stratified dummy estimator and the dashed lines the single-class dummy estimators as indicated in the legend. The top figure plots the absolute performance of all estimators while the bottom figure plots the results relative to the baseline performance (stratified dummy estimator) to show only the improvement over random. The single-class dummy estimators are only plotted where they achieved a higher score than the stratified estimator.

Table 7.4 provides the mean F-scores, standard deviation and run-time for all the evaluations highlighting the highest value in each column. The baseline performance is also repeated for convenience.

From these results it can be seen that all models outperform the selected baseline (stratified estimator) for all scores indicating a definite improvement over random.

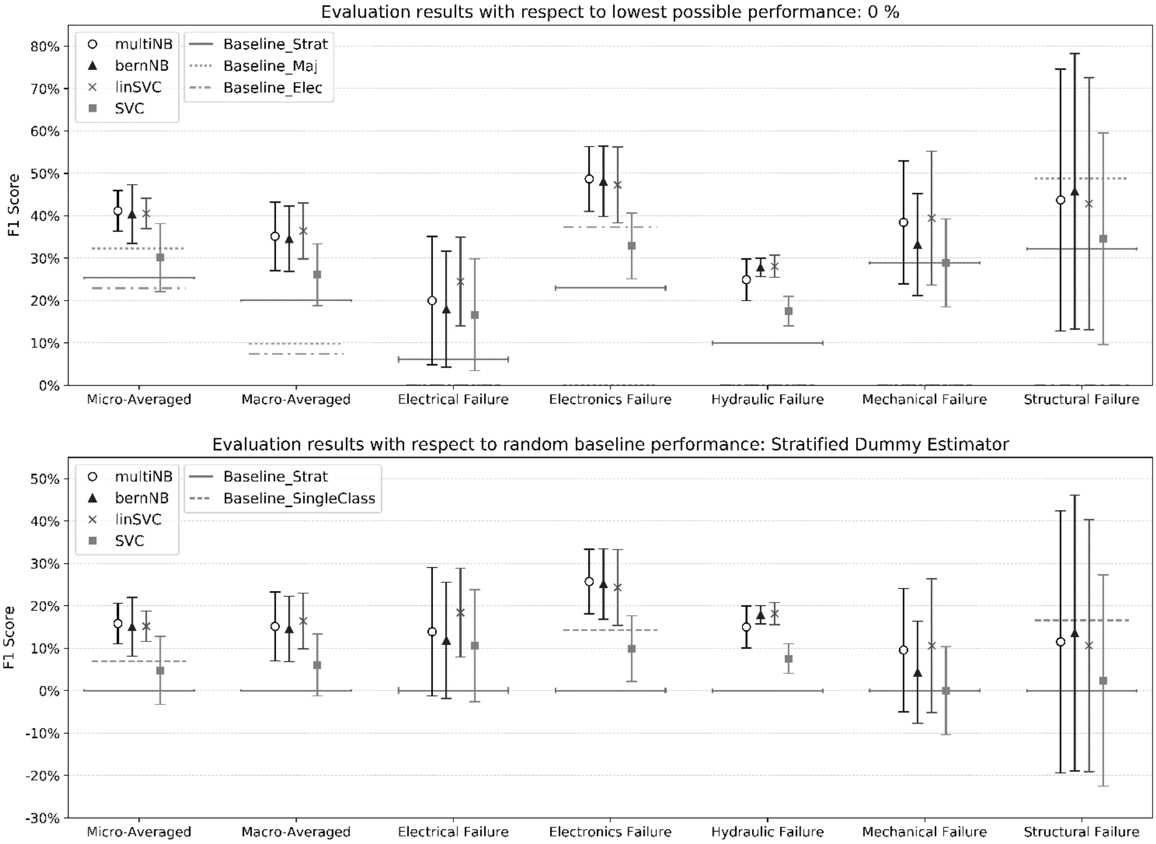


**Figure 7.2:** Nested cross-validation evaluation results

MultiNB has the highest micro-averaged score which is identified as the most important metric in Section 6.4.3 and is therefore considered the highest performing model. However, the performance difference is not substantial and both bernNB and linSVC surpass multiNB on some metrics. BernNB has the lowest running time, but does not offer a drastic advantage over multiNB or linSVC differing by less than 20 minutes. Not only is SVC the most computationally expensive method (running over 10 hours in comparison to under 2 for the rest), but it is also the consistently worst performing model. The other three models have much more comparable results for both the performance and the runtimes. While running time is not particularly constrained for this application, SVC is bordering on impractical and will only worsen if more data is collected as SVC complexity scales more than linearly with the number of documents (not features).

**Table 7.4:** Final experimental design results

| Estimator | F1-Scores (% Mean ± 1 Standard Deviation) | | | | | | | Run-Time |
|---|---|---|---|---|---|---|---|---|
| | Micro Avg. | Macro Avg. | Electrical Failure | Electronics Failure | Hydraulic Failure | Mechanical Failure | Structural Failure | (sec, min) |
| multiNB | 41,15<br>±4, 79 | 35,15<br>±8, 09 | 19,98<br>±15, 16 | 48,69<br>±7, 65 | 24,91<br>±4, 92 | 38,43<br>±14, 53 | 43,72<br>±30, 89 | 5271,03<br>(87,85) |
| bernNB | 40,38<br>±6, 92 | 34,56<br>±7, 71 | 17,93<br>±13, 68 | 48,11<br>±8, 29 | 27,81<br>±2, 17 | 33,19<br>±12, 02 | 45,77<br>±32, 54 | 4326,39<br>(72,11) |
| linSVC | 40,51<br>±3, 57 | 36,42<br>±6, 59 | 24,48<br>±10, 47 | 47,28<br>±8, 93 | 28,06<br>±2, 61 | 39,44<br>±15, 79 | 42,84<br>±29, 74 | 5142,6<br>(85,71) |
| SVC | 30,11<br>±8, 02 | 26,09<br>±7, 31 | 16,66<br>±13, 19 | 32,86<br>±7, 76 | 17,47<br>±3, 47 | 28,86<br>±10, 37 | 34,58<br>±24, 92 | 36859,13<br>(614,32) |
| Baseline_Strat | 25,34<br>±0, 27 | 20,01<br>±0, 28 | 6,09<br>±0, 79 | 22,97<br>±0, 64 | 9,92<br>±0, 58 | 28,87<br>±0, 41 | 32,21<br>±0, 37 | |
| Baseline_Maj | 32,26<br>±0 | 9,76<br>±0 | 0<br>±0 | 0<br>±0 | 0<br>±0 | 0<br>±0 | 48,78<br>±0 | |
| Baseline_Elec | 22,91<br>±0 | 7,45<br>±0 | 0<br>±0 | 37,27<br>±0 | 0<br>±0 | 0<br>±0 | 0<br>±0 | |

The models seem to vary together in terms of both their mean scores and standard deviations. For both the absolute and relative mean scores, all models perform best on Electronics Failure; but while Electrical Failure is the lowest performing class in terms of absolute scores, Mechanical Failure is the worst in terms of margin of improvement. The performance is expected to be lower for smaller classes as the models have less training data from which to learn. Since Electrical Failure is the smallest class, its low absolute performance might not be an indication of true difficulty but rather a consequence of its small size. Electronics Failure, on the other hand, is medium sized which means that it is very likely the easiest class to learn as it surpasses both Mechanical and Structural Failure which are larger. The stratified baseline model tries to account for the size advantage which is why Mechanical Failure, which has reasonably high absolute scores, has the lowest relative scores instead of Electrical Failure for which the baseline is very low. While this suggests that Mechanical Failure has the highest true difficulty (that even a human annotator might struggle with) the fact is that limited data availability, which makes Electrical Failure more difficult to learn, is an implementation reality.

The class properties also seem to affect the standard deviation of the respective per-class metrics in a fairly consistent manner with at least some correspondence to the distributional changes between folds as expected. While the variance is expected to be largest for the smallest classes, the violation of the IID assumption is also expected to increase the variance of the per-class metrics in proportion to the extent of the data-drift experienced by each class as depicted in Figure 6.2. The results reflect this to some extent as Structural Failure, which has the most drastic distributional changes from the data-drift, has the highest standard deviation for all the models despite being the largest class which should otherwise lead to more stable results. The high standard deviation of Mechanical Failure can be explained in a similar way while that of Electrical Failure, which has a relatively consistent distribution, can be explained by its small class size.

The error-bars indicate quite a large margin of performance values overall which would mean that the models exhibit relatively large levels of variance. However, these values should not be taken at face value. Because the $\mathrm{F}_{AGG}$ method only provides a single point estimate of the expected mean performance, the variance was estimated from the per-fold variation (standard deviation) around $\mathrm{F}_{AVG}$. While frequently used in literature, Forman and Scholz (2010) caution against the use of per-fold calculated F-scores (and its mean $\mathrm{F}_{AVG}$) which is sensitive to the changing class-distribution of different folds leading to inconsistent penalisation of error. This means that the reported standard deviations are likely over-estimated, and more so for some classes than others calling into question the validity of the above across-class score comparisons. This is further investigated in Section 8.4. It is important to keep this limitation in mind, but since the grouped evaluation scheme does not allow for different fold-splits, the variance could not be estimated using repeated NCV as recommended by Krstajic *et al.* (2014) leaving the per-fold scores as the only alternative.

Relative comparisons made across models, but within the same score (e.g. in Hydraulic Failure multiNB has the highest standard deviation), are more reliable as all models are evaluated on the same fold-splits and are likely to be similarly biased for each class. Then, looking only at the relative ranking of the top three models in each score in Table 7.4, it can be seen that linSVC has the lowest variance, followed by bernNB and finally multiNB which has the highest[1]. While

[1] Excluding SVC, linSVC has the lowest standard deviation for four of the seven scores including the most important: micro-averaged. BernNB has the lowest for two of the seven scores (and is lower than multiNB for four of the seven) and multiNB has the lowest for only one.

the reliability of this conclusion is also not guaranteed, it is supported by the parameter stability discussed in Section 7.1.1 and the learning curves evaluated in Section 7.4 adding more credibility to this claim.

The difference between the top three models is not substantial enough to make a definitive best selection. However, the difference between them and the lowest model SVC (in terms of both running time and performance) is enough to exclude it from further consideration.

All of the models, even the worst-performing SVC, consistently beat the stratified baseline performance with a sizeable margin (even the potentially overestimated error-bars are well beyond the range of the baseline). The top three models also beat the majority-class micro-averaged score (which is the highest baseline score for the evaluation metric) and are even able to beat the per-class performance of the single-class estimator trained for Electronics Failure; falling just short of the majority class estimator on Structural Failure. Considering the zero scores for all other classes, the learnt models are vastly superior to both single-class dummy estimators. While the actual results are not fantastic, they provide a definite improvement over all the random baselines indicating at least a level of practical value. Looking only at the margin of improvement for the micro-averaged results (most important metric) multiNB has a 15.8% increase over the stratified estimator and 8.9% over the majority-class estimator which is a sizeable improvement.

While the actual scores are far from desirable, these should be significantly improved if subject matter experts were to be involved in every step of the process and especially if better quality labelled training data were to be made available. The improvement over the random baseline for such a low quality dataset indicates that machine learning methods are at least viable for maintenance data and should be investigated further.

## 7.4 Learning Curves

Learning curves were evaluated for all the optimised models from the nested CV results except for SVC which was excluded due to its excessive training time and low performance. These curves were investigated to analyse the benefit of collecting more data as well as diagnosing the bias-variance trade-off of each model as discussed in Section 3.6. This was implemented using the Scikit-learn learning_curve module.

The learning curves are evaluated by performing cross-validation on increasing portions of the data and averaging the CV results (using $\mathrm{F}_{AGG}$). Because the curves typically start steep and become more gradual as the training data increases, the curves were evaluated for: 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the training data using smaller increases for the steepest region. Only the micro-averaged F-scores, identified as the most important metric, are reported.

The curves show the testing and the training performance as a function of the training-set size. As discussed in Section 3.6, the test (generalisation) performance tends to improve as the amount of training data increases, first drastically and then more gradually after some threshold. If the amount of training data is adequate, all models should be sufficiently beyond that threshold to where the curves start to plateau for the maximum training set size (100%). However, that does not mean more data will not be beneficial. Depending on whether the model suffers predominantly from variance or bias, it may or may not be worthwhile to collect more data.

Learning curves were evaluated for a stratified 10-fold, stratified 5-fold and grouped 5-fold CV. As for any cross-validation, increasing the number of folds will result in more training data available for each run. Therefore, using 10-folds mean that the learning curves can be extrapolated beyond the amount of training data made available to the 5-fold evaluation loop in the NCV in Section 7.3. (Stratification was required since the data contains only 5 groups preventing 10-fold grouped CV.) To ensure consistency with the evaluation strategy used in the experimental analysis, 5-fold grouped CV learning curves were also evaluated and compared with those created with stratified 10-fold CV. Finally, stratified 5-fold learning curves were also investigated to provide more insight into differences resulting from the number of folds (stratified 5-fold vs stratified 10-fold) as opposed to differences resulting from the splitting strategy (stratified 5-fold vs grouped 5-fold).

The learning curves are shown in Figure 7.3 below. The increasing training set size is indicated on the x-axis and the evaluation metric (micro-averaged F-score) is indicated on the y-axis. The top, downwards sloping curves present the training performances (re-substitution) while the bottom, upwards sloping curves indicate the testing performances (hold-out). The markers indicate the mean F-score computed using $\mathrm{F}_{AGG}$ as before. The vertical, dashed line shows the smallest training-set size used in the NCV evaluation in Section 7.3. This serves as a lower limit for those models since 4 of the 5 training folds lie further along the curve.

The first row shows the learning curves for each model evaluated using stratified 10-fold cross-validation. Because it uses 10 folds, these curves are the most complete; that is to say it is evaluated for the largest range of training set sizes. The stratified 10-fold learning curves can be evaluated up to training sets of 27 675 documents ($\frac{9}{10}$ of total data), while the stratified 5-fold maxes out at 24 600 documents ($\frac{4}{5}$ of total data). For grouped 5-fold the maximum is even smaller at 22 108 documents due to the uneven distribution of documents in groups as can be seen in Table 6.1 (limited by the smallest fold). This can clearly be seen in the second row of the figure which superimposes the learning curves for all three evaluation strategies on top of each other. (The area between the training and testing curves are shaded to make it easier to distinguish the correct pairs.) The bottom row shows only the testing curves of the three evaluation strategies. Here, the shaded area around each curve indicates the range of the per-fold test scores (connected to form per-fold test curves) showing the extent of the overlap between the different strategies. The three columns indicate the multiNB, bernNB and LinSVC models respectively.

The most important thing to notice is that all the models, for all the evaluation schemes are sufficiently beyond the initial rapid increase threshold at the smallest NCV training fold (indicated with the vertical line). This means that the pessimistic bias inherent in cross-validation estimates should not be excessive (Hastie *et al.*, 2009).

As per Section 3.6, the area between the training and testing curve is indicative of the model variance and generalisation performance. If the curves converge, the model has low variance and good generalisation performance relative to the training performance. For these models, the error is likely dominated by bias and it

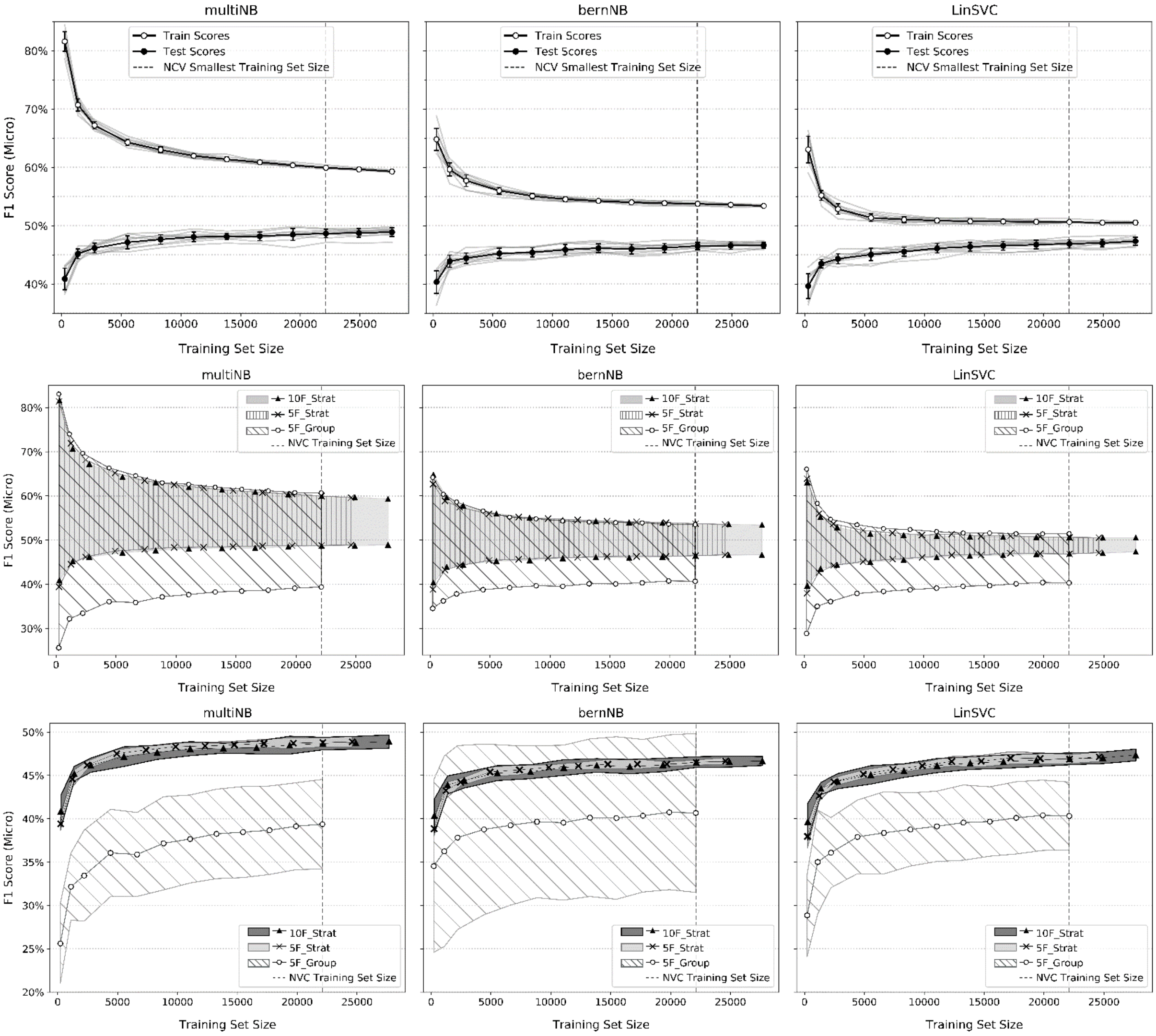


**Figure 7.3:** Learning curves of the top three models multiNB, bernNB and linSVC (from left to right). Top row: stratified 10-fold CV (faded lines showing per-fold variation). Middle row: all evaluation strategies superimposed (shaded region connecting the training and testing curves for each scheme). Bottom row: test-curves of all evaluation strategies (shaded region indicating per-fold variation).

is unlikely to benefit much from the addition of more data. Such models are likely underfitting the data and may benefit from increasing the complexity to reduce the bias (e.g. by considering more complex algorithms, more complex feature sets or reducing the regularisation).

On the other hand, models for which the curves have not yet converged suffer from high variance and are likely to benefit from the addition of more data. A much higher training performance than testing performance at the maximum sample size is indicative of poor generalisation performance relative to the training performance meaning that the model is likely overfitting the data. This can be addressed by decreasing the complexity (e.g. considering simpler algorithms, simpler feature sets or more regularisation) or by the addition of more data if the researcher feels that the complexity is warranted. Of course, high bias and high variance are not mutually exclusive, and it is possible for a model to suffer from both. A low training performance is always indicative of high bias, regardless of the curve convergence.

Looking first at the stratified 10-fold curves which are the most extensive, it can be seen that linSVC is a high-bias, low-variance model while multiNB is a low-bias, high-variance model and bernNB is somewhere in between. As discussed in Section 3.5, this inverse relationship of bias and variance is typical as reducing one often leads to increasing the other.

LinSVC has the best convergence (low variance) indicating that it is the least likely to benefit from more data. Coupled with the fact that it also has the lowest training error (high bias) suggests that the performance may be improved by increasing the model complexity. While this suggests that there may be benefit in evaluating higher order kernels (which are more complex), this would have to be implemented using LIBSVC and if SVC is any indication, the training time could become intractable (will be higher than for SVC using linear kernel) and the performance may not be in line with that of linSVC as it uses a different solver. Furthermore, several authors have demonstrated no benefit in using higher order kernels for text. Alternatively, the optimisation strategy can be modified to forcibly reduce the regularisation (controlled by parameter C) or to consider more complex pre-processing strategies (more intricate or simply more features).

What is interesting to note is that both the low variance and the high bias are supported by the evaluation results in Section 7.3 and the hyperparameter optimisation discussed in Section 7.1. In Figure 7.2 and Table 7.4 it can be seen that LinSVC has smaller standard deviations than the other models for both the micro and macro-averaged scores in support of lower variance. While the standard deviations are likely inaccurate as discussed in Section 3.8.3, the problems stemming from incompatible folds are not an issue when only compared relatively within one metric as all models are evaluated on the same fold-splits. Furthermore, unlike either of the Naïve Bayes models, linSVC did not select tri-grams in the hyper-parameter optimisation limiting features to simpler uni and bi-grams. It is also the only model that ubiquitously implemented stemming, IDF normalisation and document length normalisation all of which result in less complex feature-sets than for those selected by multiNB and bernNB which support the higher bias observed in the learning curves.

The difference between the multiNB and bernNB curves is consistent with their respective complexities. The multinomial Naïve Bayes formulation is more complex than the Bernoulli one making both the higher variance (less convergence) and the lower bias (higher training error) expected. It is not clear whether the bias or the variance dominates the bernNB model and it is likely to benefit from a reduction of both. While it has a higher variance than linSVC and should therefore be more likely to benefit from the addition of more data, its test curves plateau very fast and show a similarly low growth rate (slope) at the maximum training-set size indicating limited further potential.

Cross-validation (used by both the learning curves and the experimental design) are only an evaluation tool. Before implementation, a model will be re-trained on all the data without the need for hold-out test samples meaning a training-set size of 30 751 documents. Due to having the highest variance (least convergence), multiNB is the most likely to benefit from this increase in data. Furthermore, multiNB already has the highest testing performance at the maximum sample size of 27 675 documents and is also the highest performing model in the results from Section 7.3. This means multiNB will likely have a larger margin of superiority upon implementation making it the preferable model.

Next, the superimposed stratified 10-fold, stratified 5-fold and grouped 5-fold curves are considered to evaluate the correspondence between the different evaluation schemes. It can be seen that the 5-fold and 10-fold stratified curves are almost identical. In fact, contrary to expectations the 5-fold training curves are fractionally higher than the 10-fold curves indicating a lower pessimistic bias for 5-folds. As discussed previously, decreasing the number of folds is generally expected to increase the pessimistic bias and decrease the variance of the evaluation due to the smaller training but larger test sets (Tsamardinos *et al.*, 2015). From the range of per-fold testing curves shown in the bottom row of the figure, it can be seen that the variation of the stratified 5-fold scores is indeed less than that of the stratified 10-fold scores. However, both the expected variance reduction and the unexpected bias reduction are insignificantly small and overall the results seem to support the authors who found no significant difference between 10 and 5 folds. This suggests that the differences observed between the grouped 5-fold and stratified 10-fold curves are due to the splitting strategy and not the number of folds.

While the training curves are almost identical for all the schemes, the grouped testing curves are significantly lower than either of the stratified schemes meaning similar levels of bias, increased variance and lower generalisation performance (absolute and relative to the respective training performances). This was to be expected as the stratified schemes train and evaluate the models on artificially IID data and do not account for the changing distributions resulting from concept-drift. The grouped strategy, on the other hand, tries to account for the concept-drift by evaluating the models on chronologically out-of-sample folds.

In other words, while the actual models seem relatively unaffected by the different schemes (near identical training curves) they perform significantly worse on the grouped test because it evaluates their performance on a variable environment in comparison to the much easier stable environment used by the stratified schemes. The changing test conditions of the grouped scheme accounts not only for the lower test curves, but also for the increased variance (evident from the bigger gap between the curves) which in turn translates to larger data requirements. It is important to note that while high variance is an undesirable property, it is not so much caused by the grouped scheme as it is captured by it (while neglected by the stratified schemes) and therefore better reflects the implementation reality.

Looking at the differences between the three models there seems to be some discrepancy between the grouped and stratified schemes. For the stratified curves, the variance (indicated by the convergence) of linSVC is notably smaller than that of bernNB while on the grouped curves the variance as of linSVC and bernNB seem equivalent. Furthermore, while multiNB is the highest performing model for both stratified schemes, it is actually the lowest performing for the grouped schemes. To get a better indication of the model test behaviour, the test curves are redrawn at a better scale in the bottom of Figure 7.3 with the shaded region showing the full range of the per-fold scores.

The deviation around the means provide an alternative view of the variance and while the per-fold scores are not an ideal metric, they still contain valuable information. As expected, there is clearly much more variation in the grouped results than for either of the stratified schemes. However, looking only at the variation of the per-fold scores, linSVC once again has lower variability than bernNB in support of the stratified results. The reason for this discrepancy is not clear but may be due to the interaction of different sources of variance (training, evaluation and noise variance).

To get a better indication of the model test behaviour, the test curves are redrawn in Figure 7.4 but grouped according to scheme rather than model so that the three columns now present all the stratified 10-fold, stratified 5-fold and grouped 5-fold curves respectively. The scales are the same but the grouped plot shows a different segment of the graph to save space.

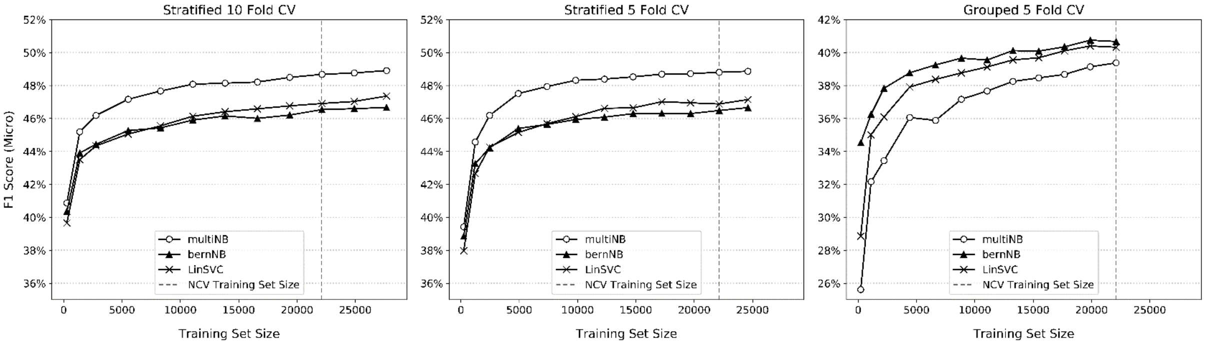


**Figure 7.4:** Test curves grouped by evaluation scheme

At this scale the difference between the stratified and grouped learning rate is much more apparent. While all the curves are sufficiently beyond the initial rapid increase phase at the smallest NCV training fold (indicated with the vertical line), the grouped curves are slower to plateau and still have significant growth rate at this point. In other words, a substantial performance increase might be possible for the grouped results if more training data was available. At the very least, the improvement should be more extensive than that observed in the stratified curves beyond the dashed line. This supports the earlier findings of higher variance that needs more data to reach a similar level of stability. It makes sense that more data is needed to properly learn the changing distributions as tested by the grouped scheme as it poses a more difficult learning task than the stratified schemes.

In contrast to the stratified results, the testing curves of the various models seem to converge for the grouped scheme. While this seems to support the findings by Mozetic *et al.* (2016) that the estimator choice is not highly significant, it is possible that if enough additional data was obtained the curves would diverge as for the stratified schemes.

The fact that multiNB is the highest performing model for both the stratified curves and for the grouped NCV evaluation in Section 7.3, but the lowest performing model on the grouped learning curves, confirms that multiNB has the greatest growth potential. The learning curves are limited by the smallest training fold. Due to the uneven distribution of documents in year-groups, most of the training folds used in the NCV analysis are significantly larger than the 22 108 documents indicated by the vertical line. This means that the results reported in Section 7.3 are for further up on the learning curves where the multiNB, which has the steepest grouped curve, has surpassed the other two curves. Furthermore, while the test scores may differ, multiNB has the highest training curves (and therefore the lowest bias) for both the grouped and stratified schemes. While it is dangerous to extrapolate beyond the observable learning curves, the results seem to indicate that future work should be focused on the Multinomial Naïve Bayes and potentially more complex variants of Support Vector Machines.

## 7.5 Comparison to Literature

Due to the substantial differences in the experimental procedure used in this paper and that reported in literature, a comparison between these results and that of the related studies evaluated in Section 4.1 is not really meaningful. Moreover according to Baharudin *et al.* (2010) such comparisons are only meaningful when comparing experiments performed by the same author in highly controlled conditions (even if the experimental procedure was the same) due to the various “background conditions” that affect the results and make such comparisons meaningless. A more useful evaluation is a methodological comparison to investigate the differences between the experimental procedures followed. This is done in the next chapter.

# Chapter 8

# Empirical Analysis: Methodological Evaluation

To validate the experimental design used in this analysis, additional experiments were performed to investigate the effects of the methodological decisions made in Chapter 6. There are some noteworthy differences between the experimental design used to obtain the results in Chapter 7, and those implemented in the related studies evaluated in Chapter 4. For some of the methodological differences, such as the use of grouped rather than stratified cross-validation, alternative methods were explicitly stated. For others, such as the cross-validated F-score computation, it is often unclear whether and how it was implemented due to inadequate documentation.

From the literature review there seems to be a significant gap between the methodological recommendations made in the more theoretical papers, and the common practice observed in more industry-focussed papers. The gap is smaller in the more general literature than for the maintenance related studies considered in Section 4.1, but even so, many papers address only one aspect of experimental validity or not at all. As per the fifth phase of the CRISP-DM methodology, it is incredibly important to evaluate not only the results, but also the process used to obtain them to ensure conservative performance estimates. This is especially important in the industrial setting where the models are used for decision-making with the potential for high safety and financial consequences. Furthermore, from

**Table 8.1:** Naming convention of experiments

| Methodological aspect | Key | Description |
|---|---|---|
| Evaluation Strategy | NCV | Nested cross-validation: outer evaluation and inner optimisation loop |
| | CV | Single loop cross validation: Same loop used for evaluation and optimisation (if any) |
| Evaluation Splitting Strategy (optimisation always stratified) | grp | Grouped (Blocked) according to year, i.e. number of splits = number of years in sample |
| | strat | Stratified (number of splits corresponding to number of years to enable comparison with grouped) |
| Optimisation | def | Default parameters, i.e. no optimisation (all others optimised) |
| More IID Subsample | iid | More IID sample selected using only data from 2007-2009. (Hence nested CV becomes 3x2 for grouped, stratified also made 3x2 to allow comparison) |
| Binary Classification (single class, one-vs-rest decomposition) | ovr_maj | OVR classifier using only the majority class: Structural Failure |
| | ovr_best | OVR classifier using only the best performing class: Electronic Failure |
| | ovr_worst | OVR classifier using only the worst performing class: Electrical Failure |
| Algorithm | multiNB | Multinomial Naïve Bayes |
| | bernNB | Bernoulli Naïve Bayes |
| | linSVC | Linear Support Vector Machine (LIBLINEAR) |
| | SVC | Support Vector Machine with linear kernel (LIBSVM) |
| Evaluation Metric | $F_{AGG}$ | Aggregated F1-score described in Section 3.8.3 (single score calculated by aggregating fold predictions) |
| | $F_{AVG}$ | Average F1-score described in Section 3.8.3 (average of per fold F1-scores) |
| | FGeom | Geometric average of the micro and macro-averaged F1-score used in parameter optimisation |

an academic perspective it is important to understand how the differences in the evaluation strategy will affect the results to enable a more fair comparison with literature.

This was investigated by performing several additional experiments on the same data changing different aspects of the experimental design to investigate their respective consequences. The micro and macro-averaged results are summarised in Figure 8.1 with the markers indicating the mean F1-score (calculated from $F_{AGG}$) and the error bars showing one standard deviation (estimated from the per-fold F1-scores) as before. Table 8.1 gives a brief description of the naming conventions used in the graph. The most important results are discussed in more

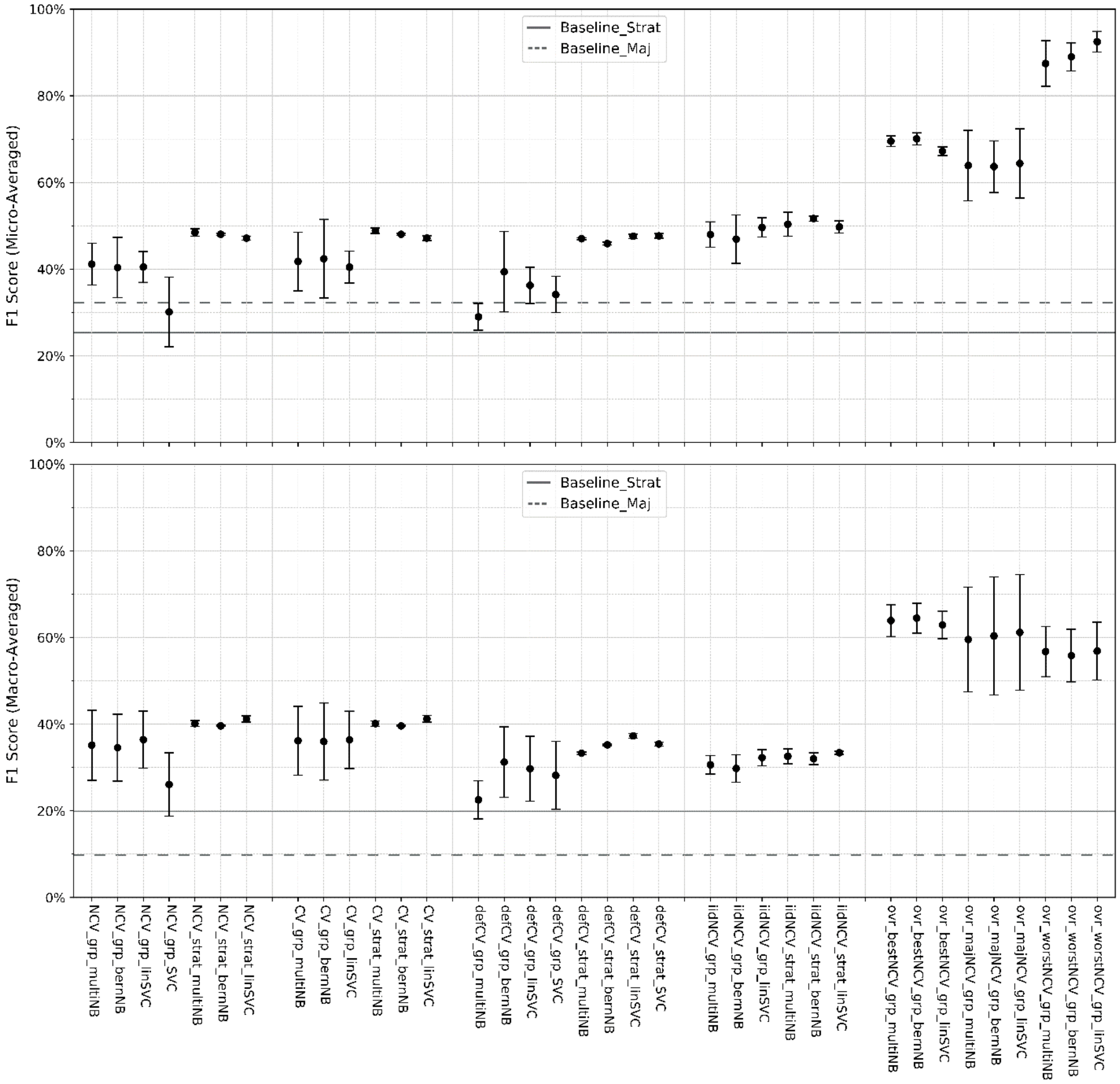


**Figure 8.1:** Results from the methodological experiments, labelled according to the naming conventions of Table 8.1

detail below. More detailed results are provided in the appendix.

## 8.1 Optimisation and Evaluation

Of the studies that did report on optimisation, none of the related studies performed it in the inner loop of a nested cross-validation. Several did not explicitly report on how (or even if) optimisation was performed while others optimised and evaluated their models on the same test-data (in either a two-way split or a single-loop CV). To see the effect of the significantly more computationally expensive NCV, the experimental procedure described in Chapter 6 was modified to a single-loop, 5-fold CV so that both the evaluation and optimisation occur in the same loop (all other parameters stay the same). In other words, the randomized-search CV is performed on the outer folds. These experiments are denoted CV (as opposed to the nested cross-validation experiments denoted NCV) with the results depicted by a grey square in Figure 8.2.

Furthermore, many of the authors did not use (or did not report on) any model optimisation and presumably used the default parameters of the implementations (or other common values reported in literature). To investigate this effect, all models were evaluated using the default parameters of both the estimator and preprocessing functions. Because no parameters are being optimised on the test sets, the inner optimisation loop of the NCV is no longer required. Accordingly, these were evaluated using a single-loop CV, but in comparison to the optimised models, the use of a single-loop is valid. These models are denoted defCV in Table 8.1 and indicated with a black triangle in Figure 8.2. The NCV results from Section 7.3 are shown with a white circle for comparison.

The running time of a single-loop CV is significantly less than for a nested one. While the NCV ran 72-88 minutes, the single-loop optimisation (which still performs 60 iterations for every fold) ran 35-48 minutes and the unoptimized, default CV ran 2-7 minutes. However, as discussed previously, the only computational limits for this task is practicality and even the SVC running time of more than 10 hours would not pose an implementation problem. Faster models are still desirable

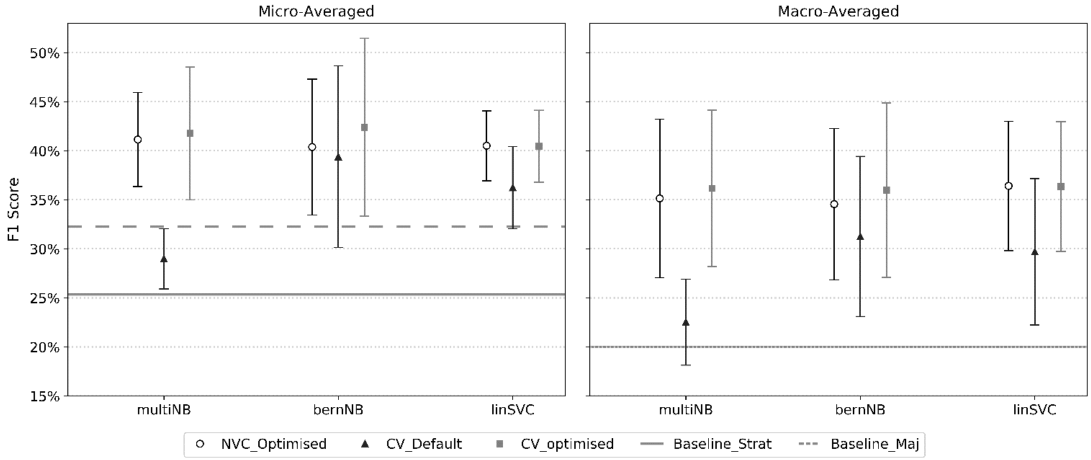


**Figure 8.2:** Optimisation strategy: Performance difference between models optimised in a NCV, a single-loop CV and unoptimised models.

for both convenience and scalability, but this is secondary to the performance and reliability of the model.

From these results it can be seen that optimisation is definitely worthwhile as it improves the results of all the models. The improvement is not only due to the refined algorithm parameters, but also due to the fact that the preprocessing transforms are customised for each algorithm. The improvement confirms the validity of a randomised search procedure in support of Bergstra and Bengio (2012) and Zheng (2015). While it is possible that an exhaustive-grid search could provide higher improvements, considering a parameter space of this size would be computationally infeasible. (Excluding the continuous parameters it would already be more than 1000 combinations for each loop as opposed to 60. Considering a reasonable range for the continuous variables it will be upwards of 100 000.)

Interestingly enough, the benefit is most substantial for multiNB, in contrast to the frequent belief that Naïve Bayes requires no optimisation. On the other hand, bernNB is the least affected, perhaps because its Boolean features limit the extent to which the preprocessing can be tuned (none of the normalisations or weighting schemes have any effect). In fact, without the optimisation, bernNB is the highest performing model. This shows the importance of comparing optimised models with each other when evaluating and selecting algorithms (Baharudin *et al.*, 2010).

Furthermore, it can be seen that the single-loop optimisation (CV) generally provides higher scores than the nested optimisation (NCV) as expected. There are two potential reasons for this difference, namely the optimistic bias of an optimised CV leading to overestimated performance (Tsamardinos *et al.*, 2015) and the pessimistic bias of NCV leading to underestimated performance[1] (Forman, 2007). However, the differences are not as big as expected and in fact, NCV actually outperforms the single-loop CV for linSVC.

Several authors have demonstrated the danger of combining the model optimization and the model assessment into a single CV as this tends to overestimate the performance. Tsamardinos *et al.* (2015) show that this effect is most significant for smaller sample sizes (<500) which may explain why the difference is not very pronounced for this dataset. They also show that the variance of NCV is significantly higher than for CV which could account for NCV surpassing CV on a few of the metrics. This effect is demonstrated by Krstajic *et al.* (2014) who show that despite NCV being lower and more reliable than CV on average, depending on the fold splits, a much wider range of NCV scores are possible (some much higher and some much lower than the corresponding CV scores) which is why they strongly recommend using repeated NCV.

Due the grouped evaluation strategy, different splits are not possible meaning that repeated NCV could not be implemented to reduce or evaluate the variance. However, it is likely that the potentially high variance of an unrepeated NCV is not so large as to invalidate the NCV results due to the sample size considered here. Tsamardinos *et al.* (2015) find that the increased variance of the NCV is a smaller concern for larger samples and already show a substantial reduction for sample sizes of 1 500. Krstajic *et al.* (2014) only considers sample sizes <5 000 which is substantially smaller than the 30 751 considered here. This seems to be supported by the error bars showing similar levels of variation between the NCV and CV results in the figure. Furthermore, Tsamardinos, et al. (2015) state that

[1] Technically both the single-loop CV and NCV are subject to a pessimistic bias as consequence of not using all of the data to train the model. However, not only is the pessimistic bias more significant for NCV, but in the single-loop CV this effect is dominated by an optimistic bias resulting from evaluating the model on the same data used to optimise it (data leakage).

the overoptimistic bias of the CV is more dangerous than the high variance of NCV.

Apart from the single CV overestimating the performance, the NCV also tends to underestimate the performance. According to Forman (2007), the optimal parameters may be sample size dependent. The single loop CV optimises parameters on the training folds (which is already smaller than the full set) while the NCV uses only half of the training folds to select parameters. This can lead to suboptimal models being evaluated and reported by the NCV leading to underestimated performances. While this is acknowledged by most of the authors recommending NCV in Section 3.8.3, it is only mentioned in passing and portrayed as inconsequential. From Tables 7.1 and 7.2, it can be seen that the parameters selected by NCV and CV differs only for multiNB and are identical for bernNB and linSVC. This supports the conclusion made from the learning curves in Section 7.4 that multiNB is most likely to benefit from more data and could account for the fact that the difference between CV and NCV is bigger for multiNB than linSVC. However, the fact that the difference between NCV and CV is larger for bernNB (which also has completely stable parameters) than for multiNB suggests that the pessimistic bias of the NCV is less significant than the optimistic bias of CV in support of Tsamardinos *et al.* (2015) and Varma and Simon (2006).

Furthermore, the learning curves in Section 7.4 also indicated that linSVC is a high bias, low variance model which is the least likely to overfit. The overestimation of the single-loop CV is due to overfitting the optimisation process to the testing folds. Because linSVC is less likely to overfit, the optimistic bias of CV is smaller so that the smaller difference between NCV and CV makes sense. This supports the notion that the difference in results are mostly due to the overestimation of CV rather than the underestimation of NCV. However, the true results are likely to be somewhere in between.

While the difference between NCV and CV is not drastic, it is enough to make bernNB the top performing model instead of the bottom which can lead to the implementation of a poorer model. Furthermore, just because the difference is small for these models on this dataset does not mean it will not be much more consequential for others (Varma and Simon (2006) demonstrated error differences of more than 20%). Unlike the decision on whether to optimise or not, this is not a question of performance but concerns the validity of the results. Therefore, while the increased computational cost (almost double) is unfortunate, it is warranted to prevent false confidence.

Once again, however, this pertains only to the calculation of performance estimates. For actual deployment the parameters would be optimised on the whole dataset using a single-loop CV and then retrained on the full dataset using the optimal parameters.

## 8.2 Splitting Strategy (Stratified vs Grouped)

None of the domain specific studies and only one of the domain independent studies used a grouped splitting strategy for cross-validation. While some of the authors did not explicitly address the splitting strategy, those that did performed either stratified or random splitting. It is generally recommended to use stratified splitting for imbalanced data such as this, with Forman and Scholz (2010) stating that stratification should always be used regardless of class imbalance.

To evaluate the effect of using the more common stratification over the chronologically grouped strategy of Bergmeir and Benitez (2012), the experimental design was repeated using stratified splitting for both the 2-fold inner optimisation loop as before, but also for the 5-fold outer evaluation loop. This means that both the optimization and evaluation was performed on artificially IID data, namely all the training and test sets have identical distributions and are also the same size. The micro and macro-averaged scores are plotted next to the grouped CV results in the left-hand column of Figure 8.3.

From these results it can be seen that the stratified approach performs consistently better than the grouped approach. Not only are the mean scores substantially higher but the spread is also significantly less. These results were expected and supports the work by Bergmeir and Benitez (2012) and Mozetic *et al.* (2018) who found that the stratified approach is likely to overestimate the performance in the presence of data drift. From Figure 6.2 in Chapter 6 it can be seen that

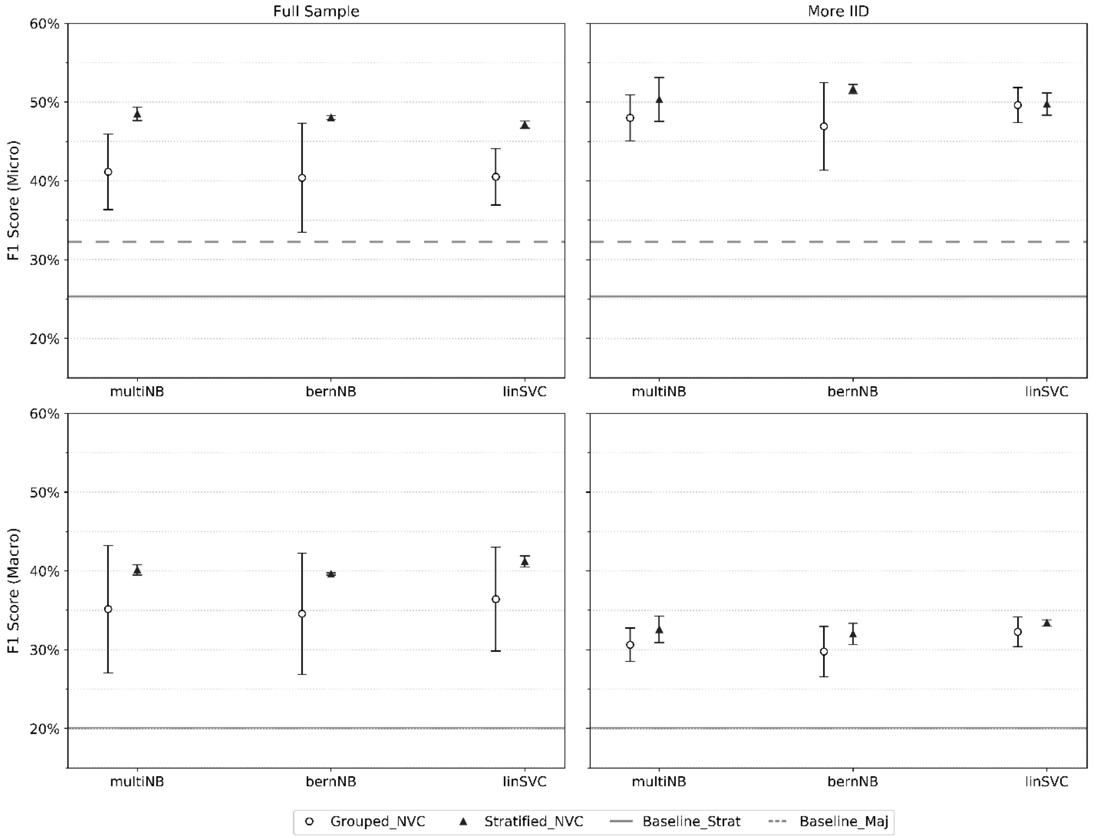


**Figure 8.3:** Different evaluation schemes (stratified vs grouped CV) for the full sample of data (left) and a more IID subsample (right)

Structural Failure and Mechanical Failure are the classes with the most extreme distributional changes between folds which should make them the most sensitive to the IID assumption of the CV. This is confirmed in Figure A.2 which shows that for all estimators, the biggest difference in mean performance for the two evaluation strategies is Structural Failure, followed by Mechanical Failure.

The difference in variance (indicated by the standard deviation error bars) was also expected due to the stratified approach artificially suppressing the variance. This is because, while the stratified CV is being applied to consistent data and therefore responds in a fairly steady manner; the grouped CV is being applied to highly irregular data and therefore responds in an inconsistent manner. Because the folds are not equally dissimilar, a wider range of results are expected as the performance is likely to vary according to the similarity between the training and the testing sets of the folds. This relates to both the model variance (stability to training data) and evaluation variance (stability of the evaluation procedure). The stratified approach seems to provide lower variance; however, this is a consequence of the artificially stable training and testing data, not due to a real reduction of variance. In other words, while the high variance exhibited by the grouped approach is undesirable, it is not caused by the grouped CV[2] but rather captured by the grouped CV (and ignored by the stratified CV). The different variances exhibited by these approaches is also confirmed by the learning curves in Section 7.4.

However, this difference is possibly exaggerated by the per-fold F-scores used to calculate the standard deviation which is likely to overestimate the variance of the grouped CV as discussed in Section 8.4. The variation is estimated from the per-fold F-scores which tend to be biased according to the degree of class-imbalance and is especially sensitive to very small classes in the test set. Forman and Scholz (2010) show that this effect is somewhat mitigated by the stratified approach which prevents the class-imbalance from becoming more extreme than the over-all dataset. This means that while the variance of the grouped approach is higher that the stratified approach as explained above, the difference might not be as extreme as indicated in the figure as the grouped CV variance is likely to be overestimated due to small class representation in some test folds. This is confirmed by Figure A.2 which shows the per class F-scores of both approaches. There it can be seen that the biggest difference exists for Structural Failure which has both the smallest (<100) and the largest (>4 500) test size for different folds as can be seen in Figure 6.2 and whose per-fold variation is therefore the most affected by the F-score bias.

To confirm that the differences observed between these two methods are due to the evaluation strategy and not indicative of a superior modelling approach, both

[2] To some extent the grouped approach does cause some additional variance as the irregular fold sizes lead to some folds with smaller test sets than those used by the stratified approach. Smaller test sets are associated with increased evaluation variance. However, those folds also have larger training sets which are associated with reduced model variance. Furthermore, other folds have this relationship reversed (larger test sets smaller training). Therefore, it is assumed that the overall effect of this is cancelled out.

the grouped and stratified experimental procedure was repeated on a sub-sample of the training set that is much closer to IID than the full sample. From the data understanding phase it could be seen that the 2007-2009 data is more homogeneous than the full set. Because this set only comprises 3 years, the evaluation was altered to a 3x2 NCV for both the stratified and grouped approach. The results are shown in the second column of Figure 8.3.

From these results it can be seen that while the stratified approach still performs better than the grouped approach, both the mean performance and the variance is much more comparable. The remaining differences make sense as the data drift was reduced but not entirely removed. This indicates that the improved score of the stratified approach is not a reflection of the true model performance but rather an unintended consequence of an erroneous evaluation strategy. In other words, while both of these approaches are valid from a model-building perspective and are likely to achieve equivalent generalisation performances in practice, only the grouped approach is valid from an evaluation perspective as the stratified approach can dangerously overestimate this performance.

Furthermore, both the stratified and the grouped micro-averaged F-scores have improved for the more IID sample despite having less training data available than their full-set counterparts. The reduced macro-averaged scores are mostly due to the extreme performance loss for Structural Failure. This is not unexpected as Structural Failure is the smallest class in the IID sample (less than half of the overall minority class: Electrical Failure) meaning that it does not have enough data to properly learn the class. Due to its small size it does not affect the micro-averaged scores much, but because all classes are weighted equally for the macro-average it has a disproportionate effect. As discussed previously, the micro-averaged results are deemed more representative of the overall model performance making the IID sample results superior. This improvement has nothing to do with the validity of the evaluation strategies, but reflects a true performance increase demonstrating the benefit of using data that conforms to the internal model assumptions as well as the increased difficulty of non-IID learning.

Both Naïve Bayes and Support Vector Machines make the IID assumption. While violating this does not affect the validity of the model (or the results) in the same manner as it does the cross-validation, they should perform better on data that matches these assumptions. Furthermore, non-IID data is likely to be indicative of a difficult task that even human annotators have difficulty with. Mozetic *et al.* (2016) showed that inter-annotator disagreement limits the theoretical performance that can be achieved by a model. While this could not be measured for this dataset, data-drift can be an indication of inter-annotator disagreement if, for example, the definition of what constitutes electrical or electronic failure changed over time, or if the failure mode assignments change according to the growing expertise of technicians. Regardless of the source of the labelling inconsistency, it poses a much more difficult learning task that requires more training data to reduce the error to acceptable levels.

This shows the value of data quality over quantity and it is likely that the performance (as measured by the grouped CV) can be substantially improved if the training data underwent a more stringent labelling process. However, the reality of the dynamic business environment is that some level of data drift is inevitable, irrespective of quality controls. Therefore it is important to have a realistic estimate of the model performance in such a changing environment as per the grouped CV. Furthermore, this also shows the value of up-to-date information which is why it is imperative to continuously monitor the model performance and update its parameters (and training data) to reflect these changes as per the CRISP-DM methodology (Chapman *et al.*, 2000).

## 8.3 Multiclass Decomposition

Most of the related studies evaluated in Section 4.1 performed binary rather than multiclass classification. This is consistent with the broader classification literature (both theoretical and practical) which is almost exclusively focused on the binary problem (Rifkin *et al.* 2003; Hoens *et al.* 2012). However, as discussed in Section 3.1, multiclass classification is inherently more difficult than binary classification and the performance is generally expected to be significantly lower (Rifkin *et al.*, 2003).

This means that the performance might be drastically improved if the problem was simplified to a binary one by identifying the most important class and combining the rest into a single "other" category. This is similar to the OVR[3] approach, except that only a single binary classifier is trained and not an ensemble. Such a model could still provide valuable information to the maintenance service provider if the most costly failure mode can be reliably detected (cost can include anything important to the service provider such as capital, production, customer experience, environmental or safety). Furthermore, this will enable a more fair comparison with the binary classification studies discussed in Section 4.1 as they cannot properly be compared to the multiclass results achieved in Section 7.3.

From the results in Section 7.3, it was observed that both the best-performing class (Electronics Failure) and the worst-performing class (Electrical Failure) are consistent for all estimators. These were selected, along with the majority class (Structural Failure), to create three one-vs-rest binary problems denoted as ovr_best, ovr_worst and ovr_maj respectively. The majority class performance is arguably the most important as it affects the largest number of documents. The best and the worst performing classes were further selected to investigate the benefit of binary classification for both an easy and a difficult class.

These were evaluated for all algorithms using the same 5x2 experimental procedure as before, except that the parameter optimisation was modified to maximise the macro-averaged F-score instead of the geometric F-score. The macro-averaged scores favour minority classes which is a desirable property for these models as the class under consideration is inevitably much smaller, but more important, than the combined "other" class. The results are shown in Figure 8.4 with the averaged metrics in the first row and only the applicable per-class scores for each OVR approach in the bottom row. (Each OVR model has only two per-class scores, the one under consideration and all others combined into “other”.) The multiclass results from Section 7.3 are also indicated for comparison. Like before, the solid horizontal line indicates the baseline scores achieved by the overall stratified dummy estimator and the dashed line spanning the micro-averaged graph shows the overall majority class dummy estimator score (not indicated for the micro-

[3] Here OVO would not be applicable as that would require knowing which of two classes all new data falls in.

averaged graph as lower than the stratified). To get a better indication of the true performance of the OVR models, additional single-class and majority class dummy estimators were evaluated for each OVR reduction to investigate how much of the model scores might be accounted for by chance.

For all the OVR models, the majority class is the combined "other" category which is used to train three additional majority-class dummy estimators (indicated as OVR_maj in graph key). These are indicated with the thin dashed line that varies according to the specific class imbalance ratio of each OVR formulation. Furthermore, a single-class dummy estimator was also trained for each of the respective class metrics (Electrical, Electronic and Structural) and indicated on those graphs with a thicker dashed line (called OVR_single in the key). The per-class scores of ovr_best and ovr_worst are identical to the overall single-class and majority class-estimators discussed in Section 7.2.

Only the majority-class OVR scores are indicated for the averaged plots as the single-class OVR scores are almost entirely below the total sample, overall baseline scores which are already substantially lower than the results achieved. By definition, both the single and majority-class OVR estimators achieve zero for the alternative class which is why they are invisible in all but one class graph.

Both the micro- and the macro-averaged scores have improved substantially for all estimators. However, from the per-class results it can be seen that this is mostly due to the high performance of the much larger, but less important "other" class. This is confirmed by the fact that the micro-scores, which favour the majority class, consistently outperform the macro-scores despite being macro-optimised.

This effect is the most evident for ovr_worst which also has the most extreme class imbalance ratio of 6:94 percent. It has both the lowest macro and highest micro scores corresponding to very low performance on the minority class of interest (Electrical Failure) and very high performance on the less important majority class ("other"). In fact, the Electrical class scores actually worsen for most of these models; though it should be noted that all of the mean scores are still safely above both the OVR single-class model (that predicts only Electrical Failure) and the overall stratified baseline.

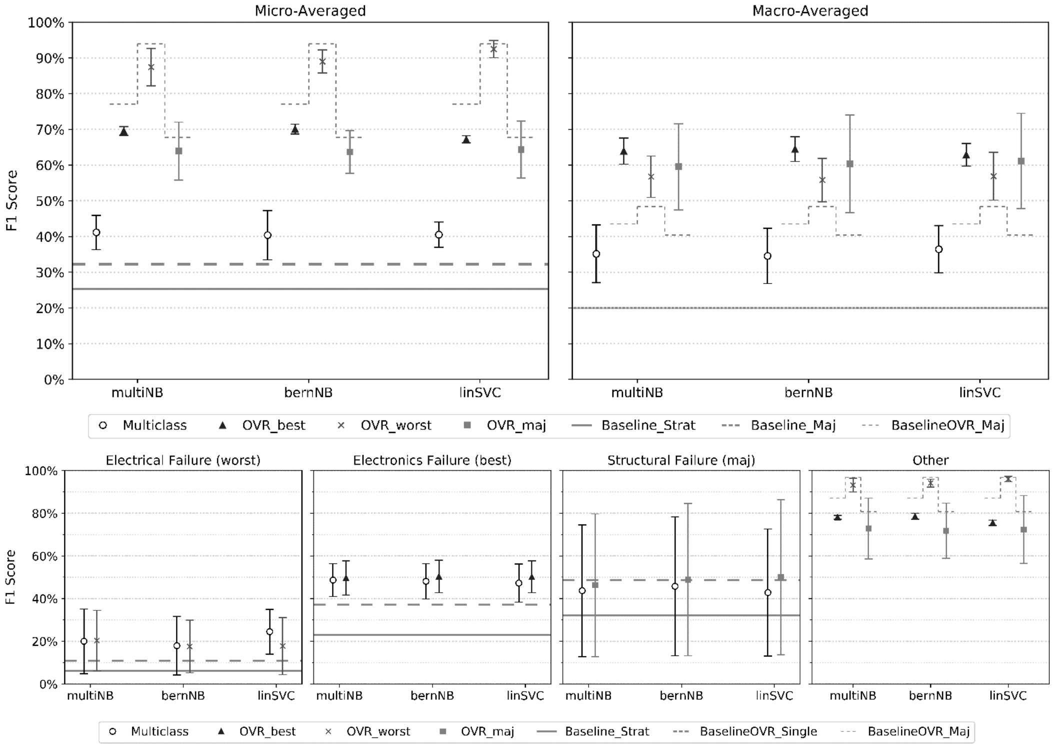


**Figure 8.4:** Micro-averaged, macro-averaged and per-class F-scores of the OVR multi-class decompositions

For ovr_best and ovr_maj the class-imbalance becomes decreasingly severe at 23:77 and 32:68 respectively. Accordingly, these models are less dominated by the majority class leading to reduced micro scores but improved macro scores corresponding to lower performance on the "other" class but higher performance on the classes of interest (Electronics and Structural Failure respectively). The biggest class improvement is for Structural Failure (ovr_maj), which is also the most balanced formulation. The Electronics class performance also increases for all ovr_best models, but the improvement is to a lesser degree. While it is still the highest scoring class, the difference between it and Structural Failure is much smaller than for the multiclass results. However, if the variance and single-class baseline results are considered, the Electronics performance is significantly better.

While none of the OVR models were able to surpass the micro-average scores achieved by always predicting "other", they all comfortably surpass the corresponding macro-averaged scores which is much more important here. While both the best and majority class scores were improved by considering them in isolation, it appears that isolating the worst class only served to improve the performance of the remaining classes. However, it is possible that this is a consequence of the class imbalance rather than an indication of the intrinsic learnability of the classes.

These results support the notion that learning becomes more difficult for increasing class-imbalance which is naturally severe for OVR implementations. It is possible that these results can be further improved by directly addressing the class imbalance using techniques such as over-sampling, under-sampling, synthetic data-generation or different cost functions. These techniques could potentially improve the results of the multiclass formulation as well as it also suffers from class imbalance. This is left for future work.

## 8.4 Metrics

Literature is frustratingly vague about many implementation details. While this is problematic from a repeatability point of view, it is also extremely challenging for new practitioners wanting to extend machine learning to different domains. One of the most important issues identified in Section 3.8.3, is the inconsistency surrounding the computation of the cross-validated F-score.

While the combination of the F-score metric with cross-validation is a widely accepted practice, there is significant discrepancy in exactly how this should be implemented. Perhaps even more dangerous, is that many authors appear unaware of this discrepancy and do not report the method used leading to incompatible comparisons made across literature (Forman and Scholz, 2010).

Forman and Scholz (2010) investigate several divergent methods found in literature and conclude that the method used in this study up to now: $\text{F}_{AGG}$, is the least biased formulation. While the theoretical proof of this claim is beyond the scope of this study, it was considered worthwhile to evaluate the difference between results generated by this method and $\text{F}_{AVG}$, the other common formulation found in literature as many of the studies considered in Chapter 4 did not specify which was used.

To investigate this effect, the per-fold F-scores of the results are plotted in the first row of Figure 8.5 showing both the $\mathrm{F}_{AGG}$ and $\mathrm{F}_{AVG}$ computed mean scores. Each fold score is indicated by a small x with the corresponding test-years indicated above in the same order. The larger markers indicate the mean scores used in all previous results: $\mathrm{F}_{AGG}$, while the small horizontal lines indicate the mean scores computed with $\mathrm{F}_{AVG}$.

Below this, the results from the stratified NCV experiment in Section 8.2 is plotted in the same way to evaluate the impact of stratified splitting on this computation. While $\mathrm{F}_{AGG}$ is always superior to $\mathrm{F}_{AVG}$ according to Forman and Scholz (2010), they do state that these results become more comparable if stratification is used to limit the class-imbalance of folds and in so doing reduce the bias of the per-fold F-scores which is the source of error for $\mathrm{F}_{AVG}$. This is important to validate the stratified hyperparameter optimisation strategy which necessitates the use of per fold F-scores in the inner loop of the NCV as these are required to compare and select different parameter combinations[4]. While these scores are not used in the performance evaluation, they can drastically affect the results by choosing suboptimal parameter combinations and should ideally be based on valid score calculations.

From the grouped CV results it can be seen that $\mathrm{F}_{AVG}$ tends to be lower than $F_{AGG}$. Comparing this tendency to the per-fold test distributions shown in Figure 6.2, it can be seen that the biggest difference between $\mathrm{F}_{AGG}$ and $\mathrm{F}_{AVG}$ is for the classes that have very small test samples in some of the folds. This consistent with the claim by Forman and Scholz (2010) that $\mathrm{F}_{AVG}$ tends to have a negative bias due to underestimating the performance of folds with very few test examples. This effect is most evident for Structural Failure whose $\mathrm{F}_{AVG}$ score is brought down significantly by the scores from 2007, 2008 and 2009; all of which have test samples below 200 documents. Removing just the fold with the smallest number

[4] The reason is two-fold. Firstly, the aggregated F-score of the inner cross-validation is the per-fold F-score of the outer cross-validation by definition. Secondly, the Scikit-learn framework uses $\mathrm{F}_{AVG}$ to evaluate cross-validation results and while this was modified for the outer loop of the NCV, changing it for the inner cross-validation embedded in the RandomizedSearchCV module is non-trivial and would no longer be an off-the-shelf implementation.

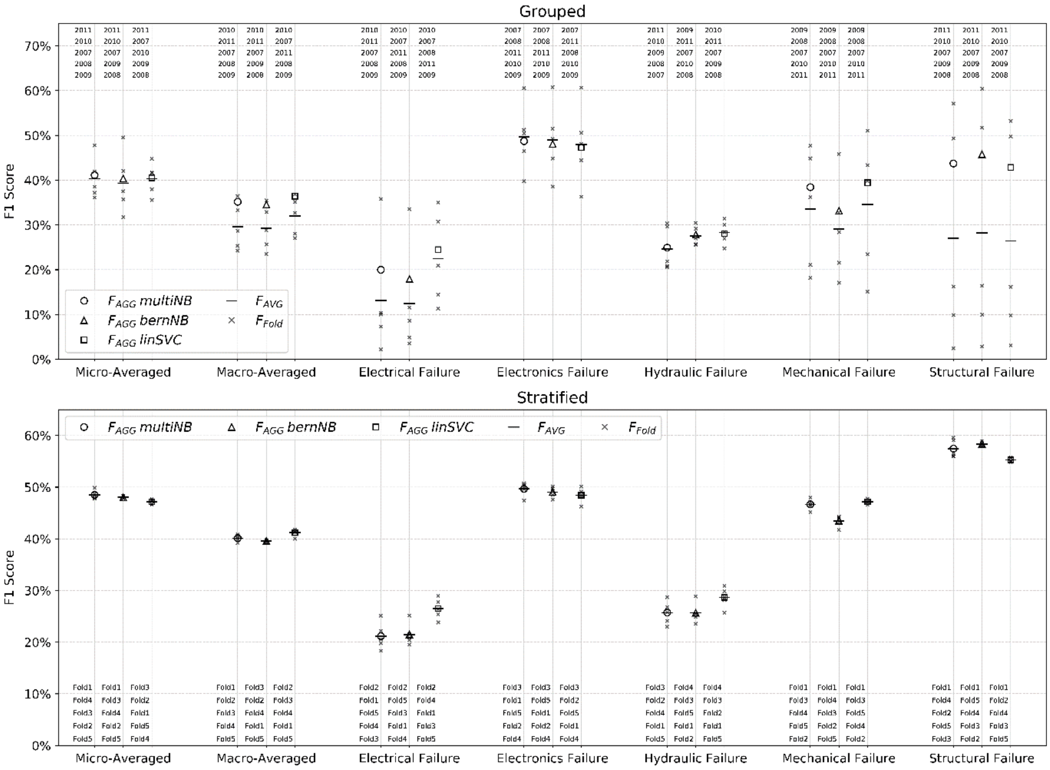


**Figure 8.5:** Difference between the aggregated F-score and the average F-score

of test documents: 2008 (which has 66 test documents) already improves the F$_{AVG}$ score by just over six percent to above 33%. In contrast, the two means are almost equivalent for Hydraulic Failure which never has below 325 documents per test fold.

According to Forman and Scholz (2010) this is due to the highly non-linear regions of the F-score function which is concave in the number of true positives (TP) and steepest near TP=0 (and undefined at zero). This means that the impact of a single test document is not fixed but is determined by the context of its test set. For instance, the cost of misclassifying a Structural Failure document in fold 2008 with 66 documents is much more significant than in fold 2011 with 4 964 documents. This is an undesirable property as it does not reflect the actual difficulty or importance of the document but rather the accidental properties of the test set. While F$_{AGG}$ is similarly curved, it avoids the highly non-linear regions near TP=0 by aggregating all the fold predictions before calculating F with a subsequently higher TP which reduces the bias significantly.

Because the model variances are estimated from the standard deviation of the per-fold scores, the error bars shown in previous figures are likely to be overestimated; especially for the classes most effected by this bias such as Structural Failure. This does not only affect the absolute values of the standard deviations but also their relative variation. Because the per-fold score bias affects some classes more than others, even the relative variation may be inaccurate. For example, if the bias could be accounted for, Structural Failure might not be the class with the highest standard deviation anymore. Unfortunately, this could not be verified as no better method of estimating the per-fold variation could be found. It should be noted that while neither the absolute value nor the relative variation across metrics is reliable, the relative variation within metrics (i.e. between models for the same score) should be fine as all models are evaluated on the same fold-splits. That means they should all be consistently biased enabling the comparison between models on a per-metric basis.

The only unexpected results was for Electronics Failure where the F$_{AVG}$ is slightly higher than F$_{AGG}$. It is also the only class where its smallest test fold, 2007, is also the highest scoring fold. While these results were unexpected and not directly accounted for by Forman and Scholz (2010), they are not entirely incompatible with their findings. The high performance of its smallest fold is not implausible as at 664 documents it is the largest minimum-test fold of all the classes and therefore the least affected by the negative bias. This makes it clear that the difference between F$_{AGG}$ and F$_{AVG}$ is not due to F$_{AVG}$ over-penalising low TP scores. However, it is less clear what the cause for this difference then is or why F$_{AVG}$ surpasses rather than equalises with F$_{AGG}$.

Forman and Scholz (2010) consider only the lower TP region of the F-score curve where the non-linearity is the most extreme to address the small test samples typically encountered in imbalanced datasets. They make no mention of the higher TP region where test-folds may differ greatly in size, but even the smallest is sufficiently beyond the highly non-linear region near TP=0 (which is the case for Electronics Failure). While the non-linearity reduces for higher values of TP, it is possible that the wide range of test-fold sizes make even minor non-linearity noticeable.

Furthermore, Forman and Scholz (2010) show that not only does $F_{AVG}$ tend to over-penalise low true positive (TP) values, but it also under-penalises high false positives (FP). From the per-fold confusion matrices it was found that Electronics Failure has a particularly high density of false positives and is the only class for which the false positives consistently outnumbers the true positives. This may suggest that for higher TP values the $F_{AVG}$ bias becomes dominated by the under-penalisation of FP rather than the over-penalisation of TP leading to a positive rather than negative bias. This positive bias would then explain $F_{AVG}$ surpassing $F_{AGG}$. However, as mentioned before, this speculation cannot be substantiated from literature and warrants further investigation in future work.

For the stratified results, there is no discernible difference between $F_{AVG}$ and $F_{AGG}$ plotted on the figure below. Whilst not exactly equal, they never differ by more than 1% and usually significantly less. These results are consistent with Forman and Scholz (2010) who found that stratification reduces the $F_{AVG}$ bias significantly. Furthermore, the per-fold scores are also much more tightly distributed than the grouped results. Unlike the grouped results, the order of the highest to lowest achieving test-folds for a class varies between the three algorithms. In the grouped results, the per-fold scores are dominated by the accidental properties of the test-folds which is why their order remains relatively constant across all algorithms in a class. The difficulty may still vary between stratified folds leading to consistently higher or lower performing folds in a class. However, the increased fluctuation of this order indicates a better response to the individual strengths and weaknesses of each estimator. Because the results are less dominated by the fold properties they also provide better reflection of the actual model performance.

This validates the use of per-fold F-scores and $F_{AVG}$ in the hyperparameter optimisation. Since the results are only ever used comparatively to select parameters and not to evaluate the models, any remaining bias will only affect the optimality of the model and not the evaluation results.

Finally, these results provide some insight into why the different F-score computation methods receive so little attention in literature. Because most studies use stratification, the difference between these methods will be very small in all but a few exceptions making the distinction between them go unnoticed. This also means that of the studies evaluated that do not specify the F-method used, only those that also used stratification can actually be compared with each other.

It should be noted that the intention of this section is not to prove the superiority of one method over another. In fact, no claim is made of the superiority of $\mathrm{F}_{AGG}$ other than citing Forman and Scholz (2010). Rather, it is to demonstrate the danger of the incompatible evaluation measures found in literature on real word data. While Forman and Scholz (2010) provide a much more detailed analysis of this subject, they used mostly synthetic data and experiments designed to isolate and maximise the differences between these measures. Therefore, it was considered worthwhile to demonstrate the non-exaggerated impact of different F-score calculations for a more realistic data analysis (where neither the data nor the experimental procedure is designed to showcase this difference).

#### 8.4.0.1 Optimisation metric: Geometric F-score

The hyperparameter-optimisation, performed in the inner loop of the NCV, requires a single metric with which to select the superior parameter set. Although the micro-averaged F-score was identified as the most important evaluation metric, it should not be favoured to the point of disregarding the minority classes during model building. To ensure more balanced estimators, the optimisation was therefore performed to maximise the geometric mean of the micro and macro scores.

Figure 8.6 plots the micro, macro and mean F-scores produced by the MultiNB optimisation runs to show the relationship between these metrics. The left-hand graph shows all 600 optimisation runs sorted according to the geometric means. It can be seen that the micro-averaged scores (indicated with the topmost, jagged blue line) are always significantly higher than the macro-averaged scores (indicated with the bottom, jagged green line). This was to be expected as the models are naturally biased towards to the majority classes (more training data) which dominates the micro-averaged scores.

The mean provides a middle ground where the documents in the minority classes are slightly upweighted by the macro scores (which weights all classes equally), but the dominance of the majority classes are maintained by the micro scores (which weights classes according to their proportionality) just to a lesser extent. The importance of this averaging effect can be seen in the figure below. While both the micro and the macro scores should ideally be maximised, they tend to react in opposite directions with high spikes in the micro-averaged scores coinciding with low spikes in the macro-averaged scores. The mean scores provide a more moderate tradeoff between these two opposing objectives.

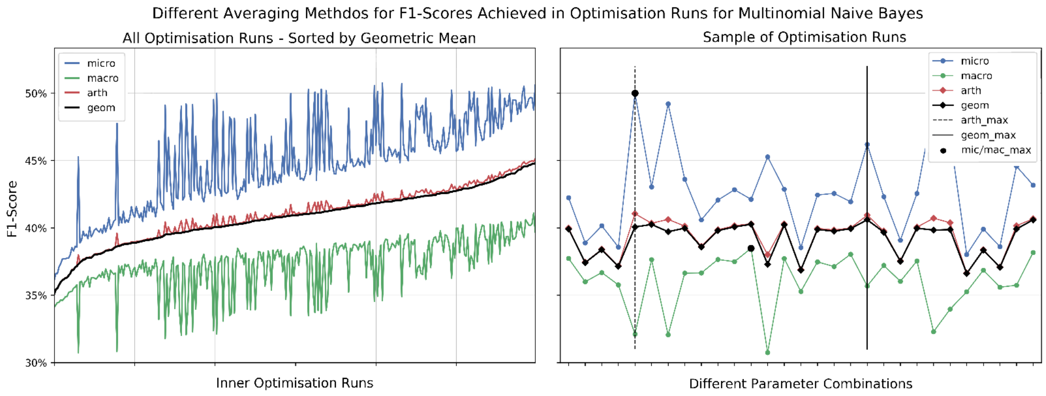


**Figure 8.6:** Difference between the optimal point selected by the geometric and arithmetic mean

Initially the arithmetic mean of the micro and macro scores was taken as the optimisation metric so that: $\mathrm{F}_{opt} = \frac{1}{2}(\mathrm{F}_{mic} + \mathrm{F}_{mac})$. However, preliminary experiments selected inconsistent parameters for the various folds. High variability in the parameter selection indicates instability of the modelling process which is undesirable from both an evaluation and implementation perspective. Furthermore, while the micro-average scores were reasonably high, the macro average scores were unacceptably low in comparison. Upon closer inspection it became apparent that the arithmetic mean was frequently dominated by extreme values, namely high micro scores corresponding to low macro scores. Therefore, the optimisation metric was changed to the geometric mean of the micro and macro averaged scores which is less sensitive to high outliers (calculated from $\sqrt{F_{mic} * F_{mac}}$ ). This led to more stable parameter sets.

Both the arithmetic and geometric means are indicated between the micro and macro scores using a red and black line respectively. From the left-hand graph it can be seen that while they follow the same over-all trends and are sometimes very near equal, the arithmetic mean (red) is more sensitive to fluctuations of the micro-averaged scores. This effect can be seen more clearly in the right-hand figure which shows the different optimum points according to the micro, macro, arithmetic and geometric means for a smaller sample of runs. The maximum arithmetic mean (indicated with a dashed, vertical line) coincides with the highest micro score (indicated by black circle on blue line) but a very low macro score. The geometric mean is also sensitive to the micro fluctuations, but in a more moderate manner. While its maximum does fall on a micro-averaged spike and a macro-average dip, both are less extreme than those of the arithmetic average.

The difference between the two mean scores are never very large, not even at their respective optimal points. However, they lead to the selection of vastly different parameter sets which can drastically affect both the evaluation results and the future implementation performance.

None of the studies considered in the literature review used the geometric mean of the micro and macro averaged scores as optimisation metric. In fact, nothing could be found in literature about taking any type of mean of the micro and macro averaged scores; for hyperparameter optimisation or otherwise. Although, since many studies omit such implementation details it is not impossible that one of them did. Regardless, the absence in literature does not pose a validity concern as it is only used in the inner optimisation loop and not for evaluation.

# Chapter 9

# Conclusion

This chapter concludes the study providing a brief overview of the project, the most important outcomes and potential areas of improvement which warrant further study and may form the basis of future research. Most importantly, it considers the completion of the project objectives and answers the research question identified in Section 1.3.

## 9.1 Project Overview

Failure data, and particularly information about failure modes, is imperative for good asset management, but frequently goes unutilised because it is buried in unstructured, natural language text which is not amenable to traditional data analytics. Several authors have acknowledged the prevalence of text-based maintenance records identifying both the potential value and the problems in utilising this data source leading many to suggest text mining as possible solution. Chapter 1 introduces both the problem area and proposed solution formulating it into the research question addressed in this study. Like Kobayashi *et al.* (2018), it identifies a gap between the academic and industry focussed text mining literature. This pertains to both the scarcity of industry (and especially maintenance specific) research and the inadequacy of the available industry and academic literature in terms of theoretical and practical considerations respectively. The most important outcome of this chapter is the overall research question and nine project objectives which seek to address this gap focussing specifically on the validity assessment which Kobayashi *et al.* (2018) identify as a critical enabler for organisational uptake.

Chapters 2-4 provide the literature review, starting with the background and motivation for the study in Chapter 2. This addresses the first two project objectives, namely, to explore the context and significance of the issues in asset management data, and to investigate the suitability of text mining to this problem and compare it with alternatives proposed in literature. The most important outcome of this chapter is the establishment of the pervasiveness of the problem in both the maintenace domain as well as the broader organisational context. Singh and Raghuvanshi (2012) report that up to 80% of all organizational data is in text format with Kobayashi *et al.* (2018) identifying text mining, and specifically text classification, as a major business opportunity.

According to Mobley (2002), one of the leading causes of ineffective asset management is the lack of factual data to support business decisions. Several authors confirm both the prevalence of, and challenges associated with, text-based maintenance records in a variety of industries including the automotive domain (Rajpathak and De, 2016), military helicopters (McKenzie *et al.*, 2010), power generation (Mukherjee and Chakraborty, 2007), railway sector (Wang *et al.*, 2017), manufacturing (Sipos *et al.*, 2014), coal mills (Uz-Zaman *et al.*, 2015), pump stations (Edwards *et al.*, 2008) and according to Mukherjee and Chakraborty (2007), Reeve (2016) and Devaney *et al.* (2005), almost all asset intensive and service organisations. Such data cannot be processed using traditional data analytics and require time-consuming and labour-intensive manual processing which few can afford. This is one of the reasons Reeve (2016) cites for up to 70% of organisations not performing even basic failure analyses on their data and Mukherjee and Chakraborty (2007) blames for the limited spread of data-driven, reliability-centred asset management forcing organisations to continue relying on intuition based, “best-guess” decision-making (Edwards *et al.*, 2008). With the exception of Reeve (2016), all the above-mentioned authors identify some form of text mining as potential solution although they acknowledge various domain specific challenges over and above the already significant difficulty of more standard text mining applications.

Reeve (2016) suggests improving the information management system according to the same principles recommended in ISO 8000 (Data Quality) and ISO 50000 (Asset Management), namely the use of coded inputs to collect structured data which enables instant search, recall and analysis. While this was recognised as the ideal solution in Chapter 2, it only improves the quality of data collected in the future whereas many companies have years of unused, potentially valuable, historical data that will then go to waste. This can be especially problematic for analyses requiring longitudinal data such as mean-time-between-failures which may be years for some equipment. Accordingly, Chapter 2 concludes the suitability of text mining to the identified problem.

Chapter 3 presents the theoretical background of both machine learning and natural language processing which are both extensively used for text mining applications. In so doing it addresses the third project objective, the outcome of which forms the theoretical framework for the subsequent chapters. One of the literature gaps identified in Chapter 1 pertains to the lack of standardisation in terms of both terminology and methodology in the text mining and machine learning literature and the need to consolidate this into a single, comprehensive framework (Moreno-Torres *et al.* 2012; Wuest *et al.* 2016). This chapter contributes towards filling this gap by sacrificing some of the broad, industry orientated scope of Kobayashi *et al.* (2018), but including several important theoretical issues such as data-drift (IID violation), the cross-validated F-score formulation and hyperparameter optimisation strategy (neglected in Kobayashi *et al.* (2018)); but in much less detail than their individual treatment in more academic papers such as Bergmeir and Benitez (2012), Forman and Scholz (2010) and Bergstra and Bengio (2012) respectively.

The final chapter of the literature review, Chapter 4, evaluates similar research done in both the maintenance-specific domain as well as the broader text classification literature to address the fourth project objective. Compared to the broader text mining literature, very few studies could be found that concern the maintenance domain, and of those that did, even fewer considered supervised and specifically multi-class classification as was the focus of this study. By far, the greatest success was achieved by those with extensive SME involvement, such as Marzec *et al.* (2014), and especially those who had access to (or created) domain specific ontologies, such as Rajpathak and De (2016), showing the extensive benefit of domain expertise. The authors without, such as Uz-Zaman *et al.* (2015), achieved much more moderate levels of success posing the question of viability for applications without these resources.

The most important outcome of this chapter, and the literature review as a whole, is the gap identified between the methods used in the industry focussed research and that recommended by the more theoretical literature of the previous chapter, especially with concern to validity assessment which is critical for industrial applications (Kobayashi *et al.*, 2018). The gap is even bigger for the maintenance specific literature, partly due to the very limited amount of research that could be found. The biggest concern pertains to the impact of the IID assumption on the validity of the various evaluation schemes and deals specifically with the use of stratification. This was not addressed in any of the domain-specific literature, despite many implicitly acknowledging the fact of its violation in their data. Other concerns regarded the choice and optimisation of preprocessing parameters and the evaluation metric used to asses performance.

Chapter 5 identifies the Cross-Industry Standard Process for Data Mining (CRISP-DM) as an appropriate methodology for this project briefly discussing its overlap with other knowledge discovery and data mining (KDDM) methodologies. Most prominent is the emphasis on iteration, context-dependency and the identification of data preparation as the most time-consuming, and critically important part of the knowledge discovery process. Both Cios *et al.* (2007) and Kurgan and Musilek (2006) identify CRISP-DM as the preferred industry model making it the ideal choice for industry-focussed research. The most important outcome of this chapter is an overview of the six phases of the CRISP-DM model which was used to guide the completion of the experimental project objectives (six to nine) as well as providing the chapter references for where each was completed.

Chapters 6-8 address the experimental project objectives according to the CRISP-DM methodology. Chapter 6 provides both the Business and Data Understanding of the particular dataset made available to this study, namely the maintenance records of one of South Africa's leading service fuel service-station brands as per the sixth project objective to use real world data. The experimental design is the most important outcome of this chapter forming the crux of the empirical analysis. Based on the outcomes of the literature review, it combines the Data Preparation, Modelling and Evaluation into a single integrated step to ensure the validity of the results. Four algorithms were considered, namely two variants of both Naïve Bayes and Support Vector Machines. Noteworthy elements of the experimental procedure include the use of:

1. *Optimised preprocessing*: treating the preprocessing steps as hyperparameters in the modelling process to prevent data leakage and to ensure optimality of the algorithm and preprocessing interaction (Krstajic *et al.*, 2014).

2. *Randomised hyperparameter optimisation*: to enable a wider search space as well as the evaluation of continuous distributions (Bergstra and Bengio, 2012).

3. *Blocked (grouped) cross validation*: to address the IID violation resulting from chronological data-drift and prevent the optimistic bias that can result from evaluating with shuffled or stratified CV (Bergmeir and Benitez, 2012).

4. *Nested Cross validation*: to separate model selection (hyperparameter optimisation) from model evaluation (blocked CV) in the inner and outer loop respectively to avoid the overoptimistic bias resulting from multiple repeated optimisations (Varma and Simon, 2006).

5. *CV Aggregated F-score*: computing the cross-validated F-score as the single metric computed from the aggregated fold predictions to prevent the bias that can result from highly non-linear edge cases (Forman and Scholz, 2010).

6. *Geometric F-score Optimisation*: performing the hyperparameter optimisation using the geometric mean of the micro- and macro-averaged F-score as optimisation metric to prevent low scores of either.

All but the last have been individually addressed in the academic literature in some level of detail and have also been implemented individually, or in combination with one or two other elements, in industry. However, while none of these elements are original, or even recent ideas, their individual utilisation remains rare in literature and no example of their combination could be found in either academic or industry-focussed research. Several papers combine one or two aspects, but, most importantly, none could be found that paired any two of NCV, aggregated F-score and blocked evaluation. Therefore, evaluating their combination, and consolidating their literature into a single review, is a valuable contribution towards the body of knowledge in not only the maintenance domain but also the broader machine learning literature. It should be noted, however, that the lack of standardised terminology makes it very possible that this exact experimental procedure is simply hidden under different names. The geometric F-score optimisation is the only exception in that no literature reference to any type of averaged micro and macro score could be found.

Chapter 7 provides the results of the experimental design implemented on the maintenance dataset in completion of both the sixth and seventh project objectives. This includes the results of the hyperparameter optimisation as well as the scores achieved by the optimised models. The optimisation results indicate relatively stable parameter selections which validate the NCV optimisation method. The most important outcome here is the somewhat surprising parameter combinations selected by the optimisation process, which differs not only from the more typical "defaults", but also between the four algorithms showing the importance of problem-specific and algorithm-specific optimisation. Along with the contradicting recommendations found in literature, this further justifies the consideration of such a large search-space as choosing a viable subset to evaluate would likely have excluded these points.

In terms of model performance, the highest micro-averaged F1-score is 41.15% and was achieved by the Multinomial Naïve Bayes which also has the highest per-class score at 48.69% for Electronics Failure. Similar results were achieved by both the Bernoulli Naïve Bayes (which was the fastest) and the LIBLINEAR implementation of of Support Vector Machines (which was the most stable) with only the LIBSVM implementation of SVM being definitively worse (in terms of both speed and performance). While the results are relatively low, they are consistently higher than the random baselines indicating at least some level of learnability for the data and objective. Considering the low quality of annotations used to train these models as well as the lack of SME involvement, the consistent improvement over the random baselines are a considerable achievement. Higher performance would need to be demonstrated to use as basis for business decision-making, but it is likely that this can be achieved if the issue of annotation quality and domain expertise is addressed.

Chapter 8 provides the methodological evaluation of the experimental design in accordance with the fifth phase of the CRISP-DM model. The purpose is not only to validate the results (objective eight), but also to assess the effect of the particular methodological elements which set it apart from the results reported in literature (objective nine). First and foremost, this analysis confirms the validity of the design decisions made in Chapter 6, showing by experiment that the motivation for their selection was warranted. Most importantly, it demonstrates the substantial difference between the stratified and blocked evaluation schemes in Section 8.2 showing the dangerously optimistic performance estimates made by stratified cross-validation applied to non-IID data resulting from data-drift.

Also significant, is both the importance of optimisation and the capability of the randomised search procedure to perform it, which is demonstrated in Section 8.1. All of the models showed substantial performance gains with the most drastic improvement by Multinomial Naïve Bayes whose performance increased by more than 12%. While possible that an exhaustive grid search would find better results, evaluating a search-space of this size would be infeasible and the substantial performance gains provided by the randomised optimisation (over the unoptimised models) demonstrate the value of this method.

The evaluation did not show a big difference between that of a single-loop optimisation and the NCV with only a slightly higher performance estimate by the single loop. While the optimistic bias of the CV seems almost negligible for this dataset, it cannot be taken to mean it will never be an issue as several authors have demonstrated significant bias on synthetic data.

It also shows the comparative difficulty of multiclass as opposed to binary performance due to both real improvement (as a result of the simplified learning objective) and the deceptive, apparent performance benefit arising from the increased class-imbalance. This is significant as the majority of maintenance literature considered binary classification.

The final important issue is the computation of the cross-validated F-score in Section 8.4 which can be computed as either the average of the per-fold F-scores ($\mathrm{F}_{AVG}$), or as a single metric from the aggregated predictions ($\mathrm{F}_{AGG}$). While this study makes no claim to the superiority of the aggregated F-score other than referencing Forman and Scholz (2010), it shows the importance of specifying the formulation demonstrating a more than 10% difference between the two formulations for some classes.

Overall, this chapter concludes that while a direct comparison with the results reported in literature is not possible due to substantial differences in the data and modelling methods used, it is evident that many of those results are potentially over-estimated due to inadequate consideration of these methodological issues.

## 9.2 Project Critique and Future Research

The limitations and delimitations discussed in Chapter 1 summarise the most important restrictions to this study and also indicate future areas of research. The abundance of unlabelled data make semi-supervised learning, excluded from the scope of this study, a potentially valuable research area.

The failure to secure industry involvement in the data analysis is likely to have severely limited the results and it would be valuable to repeat the analysis in conjunction with SMEs and on data labelled within annotation quality control schemes like that described in Mozetic *et al.* (2016) or Lewis *et al.* (2004).

No feature transformations such as Principle Component Analysis (PCA) or Latent Semantic Analysis (LSA) were applied as it was considered valuable to maintain the interpretability of the features used by the models (which is lost in these representations). While this enabled the detection of data-contamination in an earlier iteration (Section 6.2) confirming the value in this decision, these methods have shown great promise in some applications and may be worth considering in future research. Another method not considered in this study, and which the author only became aware of upon completion of the analysis, is class-specific feature selection which has been shown to greatly improve multi-class performance (Forman, 2004) and should be considered for future research.

The hyperparameter optimisation procedure was evaluated in terms of overall performance benefit and the final parameter selections, but the individual impact and behaviour of the various parameters were not considered and may be a valuable avenue of pursuit. Specifically, there may be value in modified stop-word-removal lists to, for instance, remove all negation stop-words such as *not* and *no* which has a greater than usual importance in the maintenance records (Section 6.2).

Finally it should be noted that the methodological conclusions made in this study are based on the evaluation of a single dataset and is not necessarily indicative of the general behaviour. It would be worthwhile to assess the significance of nested vs single loop cross-validation, stratified vs blocked (grouped) evaluation and the differences in the cross-validated F-score formulations on multiple datasets.

## 9.3 Project Conclusion

The answer to the research question posed in Chapter 1 is a provisional yes. Text mining (and the related fields of machine learning and natural language processing) is a viable solution for the extraction of useful information, and specifically failure modes, from the low quality, unstructured text maintenance records which are typically available in industry. Not only does the theoretical analysis confirm this, but the the potential has also been demonstrated to some extent in the empirical analysis. The reason for the provisional, rather than definitive yes, is that the potential is largely dependent on the organisation. A substantial amount of time and resources need to go into the planning, preparation, execution and maintenance of text mining systems to get the full benefit thereof, not unlike that needed for improved information management systems. Text mining does not provide a magic wand with which to fix all data quality problems, but if correctly applied it can be a powerful enabler. All the project objectives were fully completed in the study, with the exception of the ninth which could only be broadly discussed due to the large methodological differences preventing fair comparisons.

# Appendices

# Appendix A

# Additional Results

Some additional results of the experimental analysis are provided in this appendix. Table A.1 provides the per-fold class distribution in the final sample described in Section 6.2.1 to better show the distributional changes corresponding to chronological data drift. Table A.2 shows the micro-averaged, macro-averaged and per-class F1-scores of all experiments including the additional methodological experiments discussed in Chapter 8.

Figure A.1 provides the confusion matrices for the four algorithms with the rows corresponding to the Multinomial Naïve Bayes, Bernoulli Naïve Bayes, LIBLINEAR implementation of of Support Vector Machines and the LIBSVM implementation of Support Vector Machines. The left-hand column shows the raw number of documents with the diagonals showing the number of correct predictions (TP). The right-hand column is normalised according to the row-totals (true class membership) so that the diagonals provide the per-class recalls. From this the different types of error can more clearly be seen. For instance, it can be seen that Electrical Failure documents are often misclassified as Electronics Failure, a mistake that can be explained by the shared vocabulary of these two classes. Moreover, documents from all classes are frequently misclassified as Structural Failure, which is the majority class so that the mistake is driven by class-imbalance.

Finally, Figure A.2 plots the micro-averaged, macro-averaged and per-class F1-scores of the optimisation (left) and splitting strategy (right) experiments discussed in Sections 8.1 and 8.2 respectively. From this the difference in stratified and grouped cross-validation can clearly be seen for both the full sample and the more IID sub-sample.

**Table A.1:** Per-fold class distribution in selected sample

| | Testing Sets | | | | | Training Sets | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2007 | 2008 | 2009 | 2010 | 2011 | 2007 | 2008 | 2009 | 2010 | 2011 |
| Electrical Failure | 8.1% | 6.2% | 5.8% | 8.6% | 3.1% | 6.0% | 6.1% | 6.2% | 5.1% | 7.2% |
| Electronics Failure | 31.8% | 26.8% | 16.1% | 21.1% | 24.4% | 22.3% | 21.9% | 24.5% | 23.6% | 22.4% |
| Hydraulic Failure | 15.6% | 13.8% | 17.0% | 5.6% | 4.6% | 9.4% | 8.8% | 8.2% | 11.5% | 11.7% |
| Mechanical Failure | 38.9% | 52.2% | 57.9% | 11.6% | 5.7% | 28.2% | 22.8% | 22.3% | 35.6% | 37.0% |
| Structural Failure | 5.6% | 1.0% | 3.2% | 53.1% | 62.2% | 34.2% | 40.4% | 38.9% | 24.1% | 21.8% |

**Table A.2:** Micro-averaged, macro-averaged and per-class F1-scores of all experiments

| Name | | F1-Scores (% Mean ± 1 Standard Deviation) | | | | | | | | Run-Time (sec. min) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Micro Averaged | Macro Averaged | Electrical Failure | Electronics Failure | Hydraulic Failure | Mechanical Failure | Structural Failure | Other | |
| Nested CV - Optimized | NCV_grp_multiNB | 41,15 ± 4,79 | 35,15 ± 8,09 | 19,98 ± 15,16 | 48,69 ± 7,65 | 24,91 ± 4,92 | 38,43 ± 14,53 | 43,72 ± 30,89 | | 5271,03 (87,85) |
| | NCV_grp_bernNB | 40,38 ± 6,92 | 34,56 ± 7,71 | 17,93 ± 13,68 | 48,11 ± 8,29 | 27,81 ± 2,17 | 33,19 ± 12,02 | 45,77 ± 32,54 | | 4326,39 (72,11) |
| | NCV_grp_linSVC | 40,51 ± 3,57 | 36,42 ± 6,59 | 24,48 ± 10,47 | 47,28 ± 8,93 | 28,06 ± 2,61 | 39,44 ± 15,79 | 42,84 ± 29,74 | | 5142,6 (85,71) |
| | NCV_grp_SVC | 30,11 ± 8,02 | 26,09 ± 7,31 | 16,66 ± 13,19 | 32,86 ± 7,76 | 17,47 ± 3,47 | 28,86 ± 10,37 | 34,58 ± 24,92 | | 36859,13 (614,32) |
| | NCV_strat_multiNB | 48,49 ± 0,86 | 40,13 ± 0,67 | 21,14 ± 2,6 | 49,67 ± 1,31 | 25,74 ± 2,24 | 46,66 ± 1,04 | 57,46 ± 1,77 | | 5062,13 (84,37) |
| | NCV_strat_bernNB | 48,04 ± 0,23 | 39,6 ± 0,14 | 21,49 ± 2,17 | 49,03 ± 1,01 | 25,66 ± 1,96 | 43,47 ± 1,04 | 58,36 ± 0,37 | | 4607,14 (76,79) |
| | NCV_strat_linSVC | 47,15 ± 0,45 | 41,2 ± 0,7 | 26,49 ± 2 | 48,44 ± 1,42 | 28,65 ± 1,95 | 47,16 ± 0,48 | 55,27 ± 0,4 | | 5616,64 (93,61) |
| Single Loop CV - Optimized | CV_grp_multiNB | 41,77 ± 6,76 | 36,16 ± 7,97 | 19,47 ± 12,79 | 48,39 ± 8,66 | 28,56 ± 2,54 | 36,62 ± 14,4 | 47,78 ± 33,77 | | 2445,8 (40,76) |
| | CV_grp_bernNB | 42,4 ± 9,07 | 35,99 ± 8,89 | 18,89 ± 14,17 | 48,43 ± 8,15 | 28,6 ± 2,11 | 34,31 ± 12,55 | 49,7 ± 35,96 | | 2077,67 (34,63) |
| | CV_grp_linSVC | 40,45 ± 3,67 | 36,36 ± 6,62 | 24,27 ± 10,5 | 47,16 ± 8,93 | 28,1 ± 2,55 | 39,35 ± 15,74 | 42,91 ± 29,69 | | 2882,87 (48,05) |
| | CV_strat_multiNB | 48,88 ± 0,62 | 40,13 ± 0,62 | 21,39 ± 2,75 | 49,19 ± 1,13 | 24,91 ± 1,48 | 46,73 ± 0,93 | 58,42 ± 0,84 | | 2221,91 (37,03) |
| | CV_strat_bernNB | 48,04 ± 0,23 | 39,6 ± 0,14 | 21,49 ± 2,17 | 49,03 ± 1,01 | 25,66 ± 1,96 | 43,47 ± 1,04 | 58,36 ± 0,37 | | 2052,28 (34,2) |
| | CV_strat_linSVC | 47,17 ± 0,57 | 41,23 ± 0,79 | 26,27 ± 2,02 | 48,42 ± 1,34 | 28,8 ± 1,97 | 47,44 ± 0,34 | 55,23 ± 0,55 | | 3178,82 (52,98) |
| Single Loop CV - Unoptimized | defCV_grp_multiNB | 28,99 ± 3,07 | 22,53 ± 4,38 | 14,74 ± 15,39 | 45,01 ± 6,37 | 1,48 ± 2,81 | 30,27 ± 12,61 | 21,16 ± 13,54 | | 2,14 (0,04) |
| | defCV_grp_bernNB | 39,4 ± 9,26 | 31,26 ± 8,16 | 16,64 ± 16,04 | 48,02 ± 7,56 | 14,71 ± 3,46 | 32,48 ± 11,8 | 44,46 ± 32,68 | | 2,13 (0,04) |
| | defCV_grp_linSVC | 36,25 ± 4,19 | 29,72 ± 7,46 | 15,49 ± 14,12 | 44,21 ± 7,19 | 16,07 ± 9,38 | 36,43 ± 14,85 | 36,38 ± 26,25 | | 7,19 (0,12) |
| | defCV_grp_SVC | 34,14 ± 4,18 | 28,19 ± 7,83 | 14,91 ± 15,05 | 46,54 ± 7,28 | 15,7 ± 14,47 | 33,06 ± 12,78 | 30,75 ± 21,44 | | 573,5 (9,56) |
| | defCV_strat_multiNB | 47,02 ± 0,26 | 33,29 ± 0,3 | 17,15 ± 2,57 | 47 ± 0,44 | 0,72 ± 0,27 | 44,98 ± 0,54 | 56,58 ± 0,57 | | 2,35 (0,04) |
| | defCV_strat_bernNB | 45,9 ± 0,33 | 35,21 ± 0,19 | 18,08 ± 1,94 | 49,07 ± 0,95 | 14,27 ± 1,02 | 38,62 ± 0,27 | 56,01 ± 0,46 | | 2,24 (0,04) |
| | defCV_strat_linSVC | 47,58 ± 0,49 | 37,29 ± 0,58 | 18,93 ± 1,31 | 45,82 ± 0,7 | 17,19 ± 2,17 | 48,04 ± 0,51 | 56,46 ± 0,75 | | 7,18 (0,12) |
| | defCV_strat_SVC | 47,66 ± 0,54 | 35,41 ± 0,46 | 18,16 ± 2,4 | 48,72 ± 0,99 | 7,43 ± 1,34 | 46,2 ± 0,43 | 56,56 ± 0,9 | | 580,61 (9,68) |
| Nested CV - Optimized | iidNCV_grp_multiNB | 48 ± 2,91 | 30,63 ± 2,12 | 11,81 ± 5,62 | 49,7 ± 12,28 | 28,83 ± 6 | 58,17 ± 0,32 | 4,63 ± 2,37 | | 1329,57 (22,16) |
| | iidNCV_grp_bernNB | 46,92 ± 5,56 | 29,77 ± 3,2 | 11,04 ± 7,78 | 48,05 ± 9,96 | 29,99 ± 5,98 | 57,14 ± 6,13 | 2,63 ± 1,96 | | 1190,75 (19,85) |
| | iidNCV_grp_linSVC | 49,61 ± 2,22 | 32,27 ± 1,88 | 20,59 ± 6,35 | 46,69 ± 7,02 | 22,5 ± 2,94 | 62,61 ± 3,38 | 8,95 ± 2,46 | | 1351,99 (22,53) |
| | iidNCV_strat_multiNB | 50,34 ± 2,78 | 32,58 ± 1,7 | 15,66 ± 4,48 | 51,25 ± 1,12 | 27,34 ± 3,99 | 61,11 ± 4,31 | 7,53 ± 3,87 | | 1239,38 (20,66) |
| | iidNCV_strat_bernNB | 51,67 ± 0,59 | 32 ± 1,35 | 14,04 ± 5,74 | 50,44 ± 0,29 | 26,93 ± 0,84 | 63,25 ± 1,26 | 5,37 ± 1,45 | | 1094,9 (18,25) |
| | iidNCV_strat_linSVC | 49,75 ± 1,42 | 33,4 ± 0,39 | 20,82 ± 1,83 | 48,55 ± 2 | 23,31 ± 1,36 | 62,79 ± 1,23 | 11,52 ± 0,66 | | 1209,7 (20,16) |
| One-Vs-Rest, Nested CV Optimized | ovr_bestNCV_grp_multiNB | 69,51 ± 1,22 | 63,9 ± 3,67 | | 49,65 ± 8 | | | | 78,14 ± 0,89 | 4580,44 (76,34) |
| | ovr_bestNCV_grp_bernNB | 70,07 ± 1,37 | 64,46 ± 3,44 | | 50,35 ± 7,62 | | | | 78,58 ± 1,43 | 4287,64 (71,46) |
| | ovr_bestNCV_grp_linSVC | 67,2 ± 1,01 | 62,87 ± 3,16 | | 50,19 ± 7,46 | | | | 75,55 ± 1,26 | 4618,13 (76,97) |
| | ovr_majNCV_grp_multiNB | 63,9 ± 8,13 | 59,54 ± 12,05 | | | | | 46,27 ± 33,46 | 72,82 ± 14,25 | 5322,79 (88,71) |
| | ovr_majNCV_grp_bernNB | 63,66 ± 5,95 | 60,35 ± 13,65 | | | | | 48,89 ± 35,73 | 71,81 ± 12,97 | 4362,7 (72,71) |
| | ovr_majNCV_grp_linSVC | 64,38 ± 7,97 | 61,17 ± 13,33 | | | | | 50 ± 36,35 | 72,34 ± 15,9 | 4835,1 (80,58) |
| | ovr_worstNCV_grp_multiNB | 87,44 ± 5,25 | 56,76 ± 5,8 | 20,33 ± 14,19 | | | | | 93,18 ± 3,19 | 4533 (75,55) |
| | ovr_worstNCV_grp_bernNB | 88,99 ± 3,23 | 55,82 ± 6,05 | 17,54 ± 12,29 | | | | | 94,1 ± 1,84 | 4149,29 (69,15) |
| | ovr_worstNCV_grp_linSVC | 92,47 ± 2,39 | 56,88 ± 6,68 | 17,7 ± 13,38 | | | | | 96,05 ± 1,32 | 4836,69 (80,61) |
| Dummy | Baseline_Strat | 25,34 ± 0,27 | 20,01 ± 0,28 | 6,09 ± 0,79 | 22,97 ± 0,64 | 9,92 ± 0,58 | 28,87 ± 0,41 | 32,21 ± 0,37 | | |
| | Baseline_Maj | 32,26 ± 0 | 9,76 ± 0 | 0 ± 0 | 0 ± 0 | 0 ± 0 | 0 ± 0 | 48,78 ± 0 | | |
| | Baseline_Elec | 22,91 ± 0 | 7,45 ± 0 | 0 ± 0 | 37,27 ± 0 | 0 ± 0 | 0 ± 0 | 0 ± 0 | | |

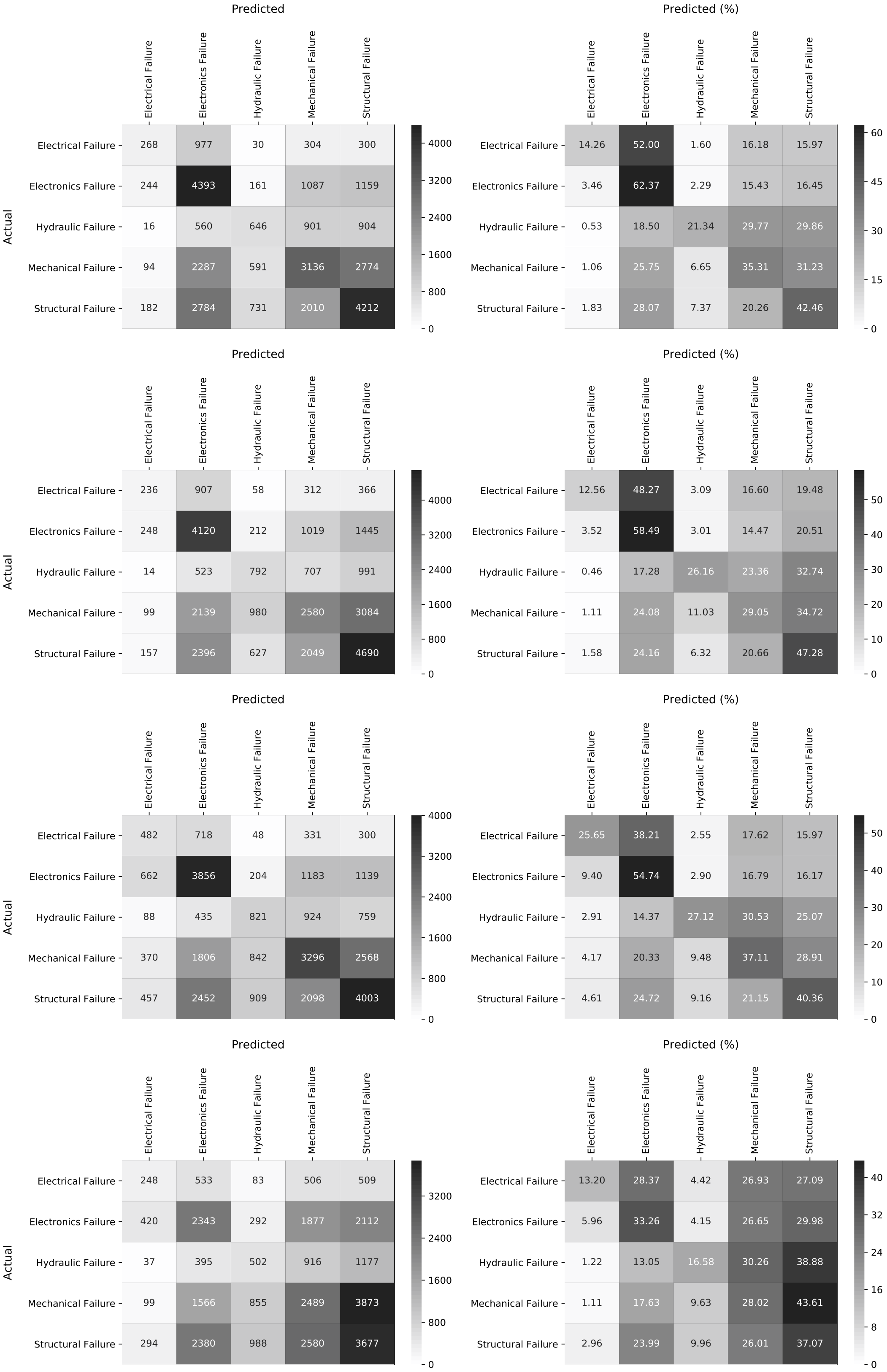


**Figure A.1:** Confusion matrices for multiNB, bernNB, linSVC and SVC

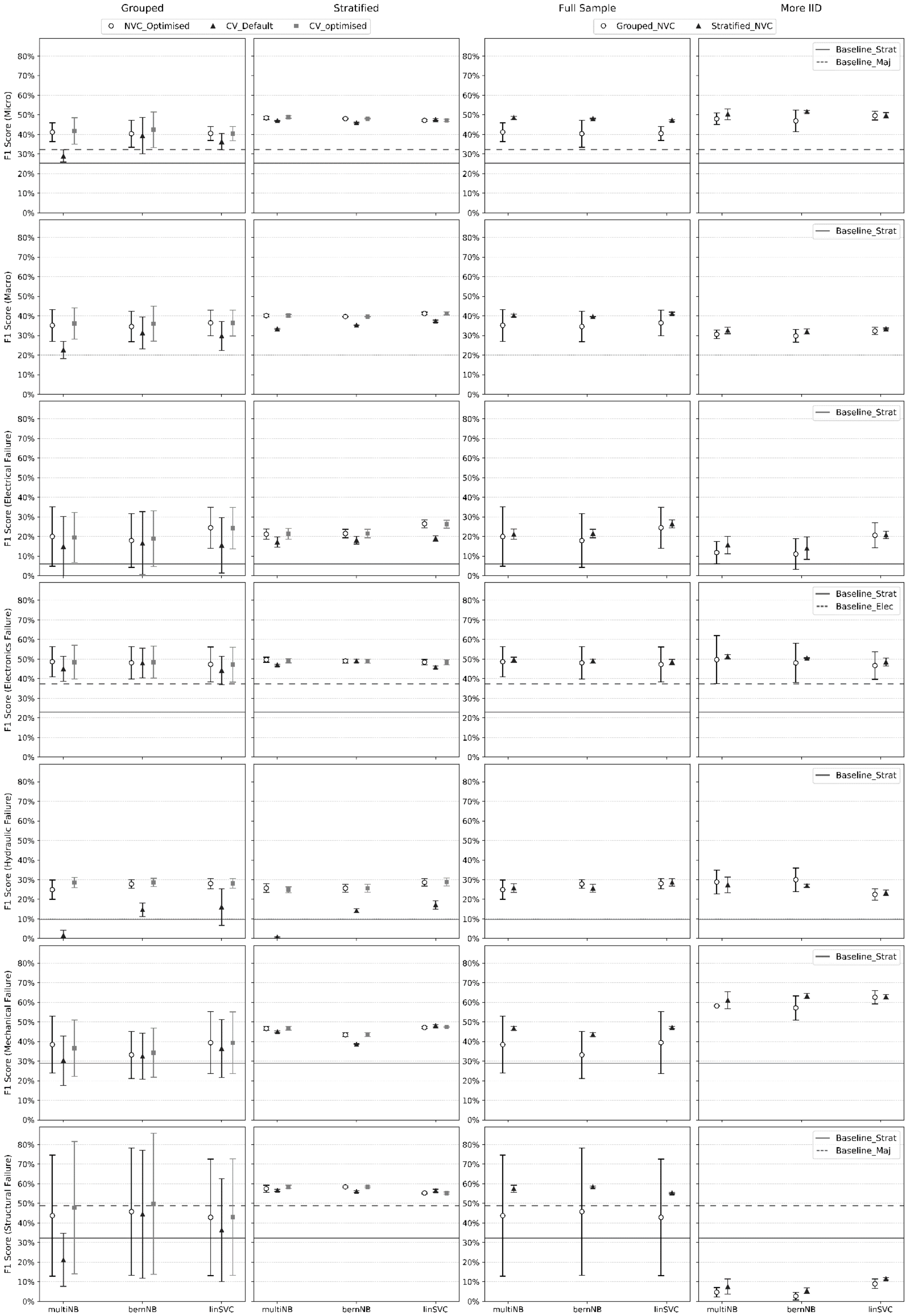


**Figure A.2:** The micro, macro and per class F-scores of the optimisation (left) and splitting strategy (right) experiments grouped by models

# List of References

Aggarwal, C.C. (2018). *Machine Learning for Text.* Springer International Publishing, New York.

Aljumaili, M. (2016). Data quality assessment: Applied in maintenance. Tech. Rep., Department of Operation, Maintenance, and Acoustics Engineering; Lulea University of Technology, Sweden.

Allahyari, M., Trippe, E., Gutierrez, J., Assef, M., Pouriyeh, S., Safaei, S. and Kochut, K. (2017). A brief survey of text mining: Classification , clustering and extraction techniques. *arXiv preprint: 1707.02919.*

Alpaydin, E. (2010). *Introduction to Machine Learning.* 2nd edn. MIT Press, Cambridge.

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L. and Ridella, S. (2012). The k in k-fold cross validation. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 441–446.

Arunraj, N.S. and Maiti, J. (2007). Risk-based maintenance - techniques and applications. *Journal of Hazardous Materials*, vol. 142, pp. 653–661.

Baglee, D., Marttonen, S. and Galar, D. (2015). The need for big data collection and analyses to support the development of an advanced maintenance strategy. *Proceedings of the International Conference on Data Mining (DMIN)*, vol. 11.

Baharudin, B., Lee, L.H. and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology*, vol. 1, pp. 4–20.

Bastos, P., Lopes, I. and Pires, L. (2014). *Safety, Reliability and Risk Analysis: Beyond the Horizon*, chap. Application of data mining in a maintenance system for failure prediction, pp. 933–940. Taylor & Francis Group, London.

Bekkerman, R. and Allan, J. (2004). Using bigrams in text categorization. Tech. Rep., Amherst.

Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C. and Popp, J. (2013). Sample size planning for classification models. *Analytica chimica acta*, vol. 760, pp. 25–33.

Bellinger, C., Sharma, S. and Japkowicz, N. (2012). One-class versus binary classification: Which and when? *International Conference on Machine Learning and Applications*, vol. 2, pp. 102–106.

Benson, P. (2008). Iso 8000 the international standard for data quality. Tech. Rep., MIT Information Quality Industry Symposium.

Bergmeir, C. and Benitez, J.M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, vol. 191, pp. 192–213.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal ofMachine Learning Research*, vol. 13, pp. 281–305.

Bermingham, A. and Smeaton, A. (2010). Classifying sentiment in microblogs: Is brevity an advantage? *Proceedings of the 19th ACM international conference on Information and knowledge management ACM*, pp. 1833–1836.

Blamey, B., Crick, T. and Oatley, G. (2012). R u :-) or :-( ? character- vs.word-gram feature selection for sentiment classification of osn corpora. *Research and Development in Intelligent Systems*, vol. 29, pp. 207–212.

Box, G. (1979). Robustness in the strategy of scientific model building. *Robustness in statistics*, pp. 201–236.

Braaksma, J., Klingenberg, W. and van Exel, P. (2011). A review of the use of asset information standards for collaboration in the process industry. *Computers in Industry*, vol. 62, pp. 337–350.

Braaksma, J. and Veldman, J. (2013). Failure mode and effect analysis in asset maintenance: a multiple case study in the process industry. *International Journal of Production Research*, vol. 51, pp. 1055–1071.

BSI (2008). Pas 55. Tech. Rep., British Standard Institute.

Carstens, W.A. (2012). Regression analysis of caterpillar 793d haul truck engine failure data and through-life diagnostic information using the proportional hazards model. Tech. Rep., Stellenbosch University, Stellenbosch.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. (2000). Crisp-dm 1.0 step-by-step data mining guide. Tech. Rep..

Chen, L. and Nayak, R. (2007). A case study of failure mode analysis with text mining methods. *Integrating Artificial Intelligence and Data Mining*, vol. 84, pp. 49–60.

Chen, M., Mao, S. and Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, vol. 19, pp. 171–209.

Chougule, R. and Chakrabarty, S. (2009). Application of ontology guided search for improved equipment diagnosis in a vehicle assembly plant. *Automation Science and Engineering*, vol. 5, pp. 90–95.

Cieslak, D.A. and Chawla, N.V. (2009). A framework for monitoring classifiers performance: when and why failure occurs? *Knowledge and Information Systems*, vol. 18, pp. 83–108.

Cios, K.J., Swiniarski, R.W., Pedrycz, W. and Kurgan, L.A. (2007). *Data Mining: A Knowledge Discovery Approach*, chap. The Knowledge Discovery Process, pp. 9–24. Springer, Boston.

Cortez, P. (2010). Data mining with neural networks and support vector machines using the r/rminer tool. *Advances in data mining. Applications and theoretical aspects*, vol. 10, pp. 572–583.

Darrell, T., Kloft, M., Pontil, M., Ratsch, G. and Rodner, E. (2015). Machine learning with interdependent and non-identically distributed data. *Dagstuhl Reports*, vol. 5, pp. 18–55.

de Vos, M., Diemer, M. and Sieborger, I. (2016). A guide to academic writing in linguistics. Tech. Rep., Rhodes.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, vol. 7, pp. 1–30.

Devaney, M., Ram, A., Qiu, H. and Lee, J. (2005). Preventing failures by mining maintenance logs with case based reasoning. In: *Proceedings of the 59th Meeting of the Society for Machinery Failure Prevention Technology*. SMFPT.

Dhanrajani, P. and Gosh, U. (2008). Mining unstructured data, a survey of text mining. *Computing for Nation Development*, vol. 2, pp. 1–5.

Dictionary, B. (2018). Failure mode.
Available at: `http://www.businessdictionary.com/definition/failure-mode.html`

Dietterich, T.G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, vol. 10, pp. 1895–1923.

Domingos, P. (1998). Occam's two razors: The sharp and the blunt. In: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 37–43. American Association for Artificial Intellegence, New York.

Domingos, P.M. (2012). A few useful things to know about machine learning. *Commun. acm*, vol. 10, pp. 78–87.

ECCMA (2019). About ISO 8000.
Available at: `https://eccma.org/iso-8000/`

Edwards, B., Zatorsky, M. and Nayak, R. (2008). Clustering and classification of maintenance logs using text data mining. *Data Mining and Analytics*, vol. 87, pp. 193–199.

Eisiminger, S. (1989). The consequences of mistranslation. *Translation Review*, vol. 30, pp. 47–49.

Forman, G. (2004). A pitfall and solution in multi-class feature selection for text classification. *Proceedings of the 21st International Conference on Machine Learning - ICML*.

Forman, G. (2007). *Computational Methods of Feature Selection*, chap. Feature Selection for Text Classification, pp. 257–277. CRC Press, Florida.

Forman, G. and Scholz, M. (2010). Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *ACM SIGKDD Explorations Newsletter*, vol. 12, pp. 49–57.

Galar, D., Kans, M. and Schmidt, B. (2015). Big data in asset management: Knowledge discovery in asset data by the means of data mining. *Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM)*, pp. 161–171.

Ganiz, M.C., Cibin, G. and Pottenger, W.M. (2010). Higher order naive bayes: A novel non-iid approach to text classification. *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1022–1034.

Gentzkow, M., Kelly, B.T. and Taddy, M. (2017). Text as data. journal of economic literature [forthcoming].
Available at: `https://web.stanford.edu/{~}gentzkow/research/text-as-data.pdf`

GFMAM (2014). *The Asset Management Landscape.* The Global Forum on Maintenance and Asset Management.

Goble, W.M. and Siebert, J.F. (2008). Field failure data – the good, the bad and the ugly. Tech. Rep., Exida, Sellersville.

Gupta, V. and Lehal, G.S. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, vol. 1, pp. 60–76.

Gutierrez, D. (2015). Text analytics: The next generation of big data.
Available at: `http://www.predictiveanalyticsworld.com/patimes/text-analytics-the-next-generation-of-big-data-061215/5529/`

Hameed, A., Khan, F. and Ahmed, S. (2014). A risk-based methodology to estimate shutdown interval considering system availability. *Process Safety Progress*, vol. 34, pp. 267–279.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning; Data Mining, Inference, and Prediction.* 2nd edn. Springer, New York.

Hipkin, I.B. and De Cock, C. (2000). Tqm and bpr: Lessons for maintenance management. *Omega*, vol. 28, pp. 277–292.

Hobbs, A. and Reason, J. (2003). *Managing Maintenance Error, a Practical Guide.* CRC Press, Florida.

Hoens, R.T., Qian, Q., Chawla, N.V. and Zhou, Z.-H. (2012). *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, chap. Building Decision Trees for the Multi-class Imbalance Problem, pp. 122–134. Springer, Berlin, Heidelberg.

Hotho, A., Nürnberger, A. and Paaß, G. (2005). A brief survey of text mining. *Ldv Forum*, vol. 20, pp. 19–62.

Hsu, C.W. and Lin, C.J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, pp. 415–425.

Hurwitz, J., Nugent, A., Halper, F. and Kaufman, M. (2013). Understanding unstructured data. In: *Big Data For Dummies*. John Wiley & Sons Inc., New Jersey.

IAM (2014). *Asset Management – An Anatomy*. The Institute of Asset Management.

Ikonomakis, M., Kotsiantis, S. and Tampakas, V. (2005). Text classification: A recent overview. *Proceedings of the 9th WSEAS International Conference on Computers, World Scientific and Engineering Academy and Society*, pp. 1–6.

ISO (2011). *ISO 8000 Data Quality Series*. International Organization for Standardization, Geneva.

ISO (2014). *ISO 55000 Asset Management Series*. International Organization for Standardization, Geneva.

Jivani, A.G. (2011). A comparative study of stemming algorithms. *Int. J. Computer Technologogy Applications*, vol. 2, pp. 1930–1938.

Kans, M. and Galar, D. (2017). The impact of maintenance 4.0 and big data analytics within strategic asset management. *Maintenance Performance Measurement and Management: conference proceedings*, vol. 6, pp. 96–103.

Karim, R., Westerberg, J., Galar, D. and Kumar, U. (2016). Maintenance analytics - the new know in maintenance. *IFAC-PapersOnLine*, vol. 49, pp. 214–219.

Kenny, C.J., Findlay, D., Thies, P.R., Shek, J. and Lazakis, I. (2017). Lessons learned from 3 years of failure: Validating an fmea with historical failure data. *European Wave and Tidal Energy Conference (EWTEC)*, vol. 12.

Kobayashi, V.B., Mol, S.T., Berkers, H.A., Kismihok, G., Hartog, D. and N, D. (2018). Text classification for organizational researchers: A tutorial. *Organizational research methods*, vol. 21, pp. 766–799.

Kostoff, R.N. (2005). Method for data and text mining and literature-based discovery. *United States Patent, Patent No. US 6,886,010 B2*.

Krstajic, D., Buturovic, L.J., Leahy, D.E. and Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, vol. 6, pp. 1–15.

Kurgan, L.A. and Musilek, P. (2006). A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, vol. 21, pp. 1–24.

Lee, C.-H. and Yang, H.-C. (2009). Construction of supervised and unsupervised learning systems for multilingual text categorization. *Expert Systems with Applications*, vol. 36, pp. 2400–2410.

Leopold, E. and Kindermann, J. (2002). Text categorization with support vector machines. How to represent texts in input space? *Machine Learning*, vol. 46, pp. 423–444.

Lewis, D.D., Yang, Y., Rose, T.G. and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, vol. 5, pp. 361–397.

Luz, A. (2017). Why you should be plotting learning curves in your next machine learning project.
Available at: `https://medium.com/@datalesdatales/why-you-should-be-plotting-learning-curves-in-your-next-machine-learning-project-221bae60c53`

Mahmood, M., Keast, R. and Brown, K. (2015). Asset management capability maturity model. Tech. Rep., Asset Institute, Queensland, Australia.

Maimon, O. and Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*. 2nd edn. Springer, New York.

Marquez, A.C. (2007). *The Maintenance Management Framework: Models and Methods for Complex Systems Maintenance*. Springer.

Marzec, M., Uhl, T. and Michalak, D. (2014). Verification of text mining techniques accuracy when dealing with urban buses maintenance data. *Diagnostyka*, vol. 15, pp. 51–57.

McAfee, A. and Brynjolfsson, E. (2012 2012). Big data: The management revolution. *Harvard Business Review*, pp. 60–70.

Mccallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752.

McCallum, E. (2012). *Bad Data Handbook: Cleaning Up The Data So You Can Get Back To Work*. 1st edn. O'Reilly Media, Sebastopol.

McKenzie, A., Matthews, M., Goodman, N. and Bayoumi, A. (2010). Information extraction from helicopter maintenance records as a springboard for the future of maintenance text analysis. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 590–600.

Mertsalov, K. and McCreary, M. (2009). Document classification with support vector machines. Tech. Rep., New York.

Mobley, K.R. (2002). *An introduction to Predictive Maintenance*. 2nd edn. Butterworth-Heinemann (Elsevier), Woburn.

Moreno-Torres, J.G., Raeder, T., Alaiz-Rodriguez, R., Chawla, N.V. and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, vol. 45, pp. 521–530.

Moubray, J. (1997). *Reliability centered maintenance*. 2nd edn. Industrial Press Inc., New York.

Mozetic, I., Grčar, M. and Smailovic, J. (2016). Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, vol. 11, pp. 1–26.

Mozetic, I., Torgo, L., Cerqueira, V. and Smailovic, J. (2018). How to evaluate sentiment classifiers for twitter time-ordered data? *PLoS ONE*, vol. 13, pp. 1–22.

Mukherjee, S. and Chakraborty, A. (2007). Automated fault tree generation: bridging reliability with text mining. *Reliability and Maintainability Symposium*, pp. 83–88.

Nowak, R.D., Singh, A. and Zhu, X. (2009). Unlabeled data: Now it helps, now it doesn't. *Advances in neural information processing systems*, pp. 1513–1520.

Palmer, R.D. (2006). *Maintenance planning and scheduling handbook*. McGraw-Hill, United States.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.

Penrose, H.W. (2008). *Physical Asset Management for the Executive*. Success by Design, Old Saybrook.

Perez, S. (2018). Twitter's doubling of character count from 140 to 280 had little impact on length of tweets.
Available at: `https://techcrunch.com/2018/10/30/twitters-doubling-of-character-count-from-140-to-280-had-little-impact-on-length-of-tweets/`

Piatetsky, G. (2018). Top software for analytics, data science, machine learning in 2018: Trends and analysis.
Available at: `https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html`

Plous, S. (1992). *The Psychology of Judgment and Decision Making*. McGraw-Hill Education, USA.

Poli, R. (2003). Descriptive, formal and formalised ontologies. *Husserl's Logical Investigations Reconsidered*, vol. 48, pp. 183–210.

Press, O.U. (2011). The OEC: Facts about the language.
Available at: `https://web.archive.org/web/20120104170705/http://oxforddictionaries.com:80/words/the-oec-facts-about-the-language`

Rajpathak, D. and Chougule, R. (2011). A generic ontology development framework for data integration and decision support in a distributed environment. *International Journal of Computer Integrated Manufacturing*, vol. 24, pp. 154–170.

Rajpathak, D. and De, S. (2016). A data and ontology driven text mining based construction of reliability model to analyze and predict component failures. *Knowledge and Information Systems,*, vol. 46, pp. 87–113.

Rajpathak, D., Siva Subramania, H. and Bandyopadhyay, P. (2012). Ontology-driven data collection and validation framework for the diagnosis of vehicle healthmanagement. *International Journal of Computer Integrated Manufacturing*, vol. 25, pp. 774–789.

Rajpathak, D.G. (2013). An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain. *Computers in Industry*, vol. 64, pp. 565–580.

Raschka, S. (2015). *Python Machine Learning.* PACKT, Mumbai.

Reese, R.M., Reese, J.L., Kaluza, B., Kamath, U. and Choppella, K. (2017). *Machine Learning: End-to-End guide for Java developers: Data Analysis, Machine Learning, and Neural Networks simplified.* Packt Publishing Ltd, Birmingham.

Reeve, J. (2016 2016). Without accurate failure data, all you have is a work order ticket system. *Uptime*, pp. 26–30.

Rennie, J.D.M., Shih, L., Teevan, J. and Karger, D.R. (2003). Tackling the poor assumptions of naive bayes text classifiers. *Proceedings of the Twentieth International Conference on Machine Learning (icml-03)*, pp. 616–623.

Rifkin, R., Mukherjee, S., Tamayo, P., Ramaswamy, S., Yeang, C.-H., Angelo, M., Reich, M., Poggio, T., Lander, E.S., Golub, T.R. and Mesirov, J.P. (2003). An analytical method for multiclass molecular cancer classification. *Siam Review*, vol. 45, pp. 706–723.

Ritchi, N. (2019). Learning curve.
Available at: `https://www.ritchieng.com/machinelearning-learning-curve/`

Rodriguez, J.D., Perez, A. and Lozano, J.A. (2010). Sensitivity analysis of k-fold cross validation. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 32, pp. 569–575.

Rudov-Clark, S.D. and Stecki, J. (2009). The language of FMEA: on the effective use and reuse of fmea. *DSTO International Conference on Health & Usage Monitoring*, vol. 6, pp. 9–12.

Russom, P. (2011 2011). Big data analytics. *TDWI Best Practices Report*, pp. 1–35.

Salzberg, S. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, vol. 1, pp. 317–328.

Samuel, A.L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, vol. 3, pp. 210–229.

Samuel, M.P., Mukhopadhyay, C. and Shankar, V. (2006). Failure mode identification and data preparation for aeroengine reliability studies. *International Seminar on Fatigue, Fracture and Durability & Symposium on Residual Life Assessment and Extension of Ageing Structures*, pp. 1–8.

Santafe, G., Inza, I.n. and Lozano, J.A. (2015). Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review*, vol. 44, pp. 467–508.

Sapkota, U., Solorio, T., Montes, M. and Bethard, S. (2015). Not all character n-grams are created equal: A study in authorship attribution. *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 93–102.

Sathi, A. (2012). *Big Data Analytics: Disruptive Technologies for Changing the Game.* 2nd edn. MC Press, Boise.

Schneider, K.-m. (2005). Techniques for improving the performance of naive bayes for text classification. *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 682–693.

Schuld, M., Sinayskiy, I. and Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, vol. 56, pp. 172–185.

Science, T. (2016). Risks & choices - background information a scientific view of risk. Available at: `http://www.terrificscience.org/lessonpdfs/Scientific_View_of_Risk.pdf`

Serrano, M.A., Flammini, A. and Menczer, F. (2009). Modeling statistical properties of written text. *Public Library of Science (PLoS ONE)*, vol. 4, pp. 1–9.

Sharma, A. and Yadava, G.S. (2011). A literature review and future perspectives on maintenance optimization. *Journal of Quality in Maintenance Engineering*, vol. 17, pp. 5–25.

Sharma, S. (2008). An integrated knowledge discovery and data mining process model. Tech. Rep., Virginia Commonwealth University, Virginia.

Sikorska, J.Z. and Hodkiewicz, L.M. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, vol. 25, pp. 1803–1836.

Singh, P.D. and Raghuvanshi, J. (2012). Rising of text mining technique: As unforeseen-part of data mining. *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 1, pp. 139–144.

Sinoara, R.A., João, A. and Solange, O.R. (2017). Text mining and semantics: a systematic mapping study. *Journal of the Brazilian Computer Society*, vol. 23, p. 9.

Sipos, R., Fradkin, D., Moerchen, F. and Wang, Z. (2014). Log-based predictive maintenance. *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1867–1876.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, vol. 45, pp. 427–437.

Stapor, K. (2018). Evaluating and comparing classifiers: Review, some recommendations and limitations. *Advances in Intelligent Systems and Computing*, vol. 578, pp. 12–21.

StatisticsHowTo (2016). IID statistics and random sampling. Available at: `https://www.statisticshowto.datasciencecentral.com/iid-statistics/`

Stolcke, A., Kajarekar, S. and Ferrer, L. (2008). Nonparametric feature normalization for SVM-based speaker verification. *Acoustics, Speech and Signal Processing*, pp. 1577–1580.

Syeda, K.N., Shirazi, S.N., Naqvi, S.A.A., Parkinson, H.J. and Bamford, G. (2018). Big data and natural language processing for analysing railway safety. *Innovative Applications of Big Data in the Railway Industry, IGI Global*, pp. 240–267.

Tan, A.-H. (1999). Text mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, vol. 8, pp. 65–70.

Tan, C.-m., Wang, Y.-f. and Lee, C.-d. (2002). The use of bigrams to enhance text categorization. *Information processing & management*, vol. 38, pp. 529–546.

Timonen, M. (2012). Categorization of very short documents. *International Conference on Knowledge Discovery and Information Retrieval*, pp. 5–16.

Tsamardinos, I., Rakhshani, A. and Lagani, V. (2015). Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. *International Journal on Artificial Intelligence Tools*, vol. 24, pp. 1–30.

Tumer, I.Y., Stone, R.B. and Bell, D.G. (2003). Requirements for a failure mode taxonomy for use in conceptual design. *Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm*, vol. 3, pp. 1–11.

Tversky, A. and Kahneman, D. (1973). Judgement under uncertainty: Heuristics and biases. Tech. Rep., Oregon Research Institute, Oregon.

Uz-Zaman, K.A., Cholette, M.E., Li, F., Ma, L. and Karim, A. (2015). A data fusion approach of multiple maintenance data sources for real-world reliability modelling. *Proceedings of the 10th World Congress on Engineering Asset Management*, pp. 69–77.

Varma, S. and Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, vol. 7, pp. 1–9.

Wachsmuth, H. (2015). *Text Analysis Pipelines: Towards Ad-hoc Large-Scale Text Mining*, chap. Text Analysis Pipelines, pp. 19–53. Springer, Weimar.

Wang, F., Xu, T., Tang, T., Zhou, M. and Wang, H. (2017). Bilevel feature extraction-based text mining for fault diagnosis of railway systems. *IEEE Transactions on Intelligent Transportation Systems,*, vol. 18, pp. 49–58.

Wang, S. and Manning, C.D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 90–94.

Welte, T.M. and Wang, K. (2014). Models for lifetime estimation: an overview with focus on applications to wind turbines. *Advances in Manufacturing*, vol. 2, pp. 79–87.

Wilbur, W.J. and Kim, W. (2009). The ineffectiveness of within-document term frequency in text classification. *Information retrieval*, vol. 12, pp. 509–525.

Witten, I.H., Frank, E. and Hall, M.A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd edn. Elsevier, Burlington.

Wolpert, D.H. (1996). The lack of a priori distinctions between learning. *Neural Computation*, vol. 8, pp. 1341–1390.

Woodall, P., Gao, J., Parlikad, A. and Koronios, A. (2015). *Engineering Asset Management-Systems, Professional Practices and Certification*, chap. Classifying Data Quality Problems in Asset Management, pp. 321–334. Springer, Cham.

Wuest, T., Weimer, D., Irgens, C. and Thoben, K.D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production and Manufacturing Research*, vol. 4, pp. 23–45.

Zentgraf, D.C. (2015). What every programmer absolutely, positively needs to know about encodings and character sets to work with text.
Available at: `http://kunststube.net/encoding/`

Zhang, X.Y., Wang, S. and Yun, X. (2015). Bidirectional active learning: A two-way exploration into unlabeled and labeled data set. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 3034–3044.

Zheng, A. (2015). *Evaluating Machine Learning Models: A Beginner's Guide to Key Concepts and Pitfalls*. O'Reilly Media, Inc., Sebastopol.

Zhu, X. and Goldberg, A.B. (2009). *Introduction to semi-supervised learning*. Morgan & Claypool Publishers, San Rafael.

Zimak, D.A. (2006). Algorithms and analysis for multi-category classification. Tech. Rep., Urbana.

Zschech, P. (2018). A taxonomy of recurring data analysis problems in maintenance analytics. *European Conference on Information Systems*, vol. 26, pp. 1–16.