Application of statistics and machine learning in healthcare

Schalk Gerhardus van der Merwe

Report presented in partial fulfilment
of the requirements for the degree of
MCom **(Mathematical statistics**)
at the University of Stellenbosch

Supervisor: Dr Chris Muller

Degree of confidentiality: A

April 2019

PLAGIARISM DECLARATION

1. Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.

2. I agree that plagiarism is a punishable offence because it constitutes theft.

3. I also understand that direct translations are plagiarism.

4. Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.

5. I declare that the work contained in this assignment, except otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

SG van der Merwe	April 2019
Initials and surname	**Date**

Acknowledgements

Dr Chris Muller – Supervisor

Jannie van Schalkwyk – Head of department, Analytics and Reporting, Mediclinic International

Cindy Morgan – Statistical analyst manager, Analytics and Reporting, Mediclinic International

Abstract

Key words: Healthcare, machine learning, patients, readmissions, statistical models

Clinical performance and cost efficiency are key focus areas in the healthcare industry, since providing quality and affordable healthcare is a continuing challenge. The goal of this research is to use statistical analyses and modelling to improve efficiency in healthcare by focussing on readmissions. Patients readmitted to hospital can indicate poor clinical care and have immense cost implications. It is advantageous if readmissions can be kept to a minimum.

Generally, stakeholders view strategies to address the clinical performance of healthcare providers, such as readmission rate, as mainly clinical in nature. However, this study will investigate the potential role of machine learning in the improvement of clinical outcomes. This study defines machine learning as the identification of complex patterns (linear or non – linear) present in observed data, with the goal of predicting a certain outcome for new cases by mimicking the true underlying pattern in the population which led to the observed outcomes in the sample while throughout limiting rigid structural assumptions.

The question at hand is whether patients that are at risk of readmission can be identified, along with the risk factors that can be associated with an increase in the likelihood of the event of readmission occurring. If yes, this can provide an opportunity to reduce the number of readmissions and thus avoid the resulting cost and clinical consequences. Once identified as a patient at risk for readmission, it will provide an opportunity for early clinical intervention. In addition, the model will provide the opportunity to calculate risk scores for patients, which in turn will enable risk adjustment of the readmissions rates reported.

The data under consideration in this study is healthcare data generated by the operations of an international healthcare provider, Mediclinic International. The data that the research is based on is patient data captured on hospital level in all Mediclinic hospitals, operational in Mediclinic International's Southern African platform.

Several statistical algorithms exist to model the responses of interest. The techniques consist of simple, well known techniques, as well as techniques that are more advanced. Logistic regression and decision trees are examples of simple techniques, while neural networks and support vector machines (SVM) are more complex. SAS Enterprise Guide is the software of choice for the data preparation, while SAS Enterprise Miner is the software used for the machine learning component of this study. The study aims to provide insight into machine learning techniques, as well as construct machine learning models that produce reasonable accuracy in terms of prediction of readmissions.

Opsomming

Sleutelwoorde: Gesondheidsorg, statistiese leer teorie, pasiënte, hertoelatings, statistiese modelle

In die privaat gesondsheidsorg industrie word daar klem gelê op meting van kliniese prestasie en koste doeltreffendheid, weens die feit dat die lewering van kwaliteit en bekostigbare gesondheidssorg ´n voortslepende uitdaging is. Die doel van hierdie studie is om statistiese analises te beskou wat die potensiaal het om ´n bydrae te lewer tot die taak om doeltreffendheid in gesondheidsorg te verbeter. Die studie beskou hoofsaaklik hertoelatings weens die belangrikheid van hertoelatings as ´n maatstaf van die kwaliteit van gesondheidssorg asook as gevolg van die onmeentlike finansiële gevolge wat hertoelatings teweeg bring. Die voordele verbonde aan die vermindering van die aantal hertoelatings, is merkwaardig.

Oor die algemeen beskou belanghebbendes die strategieë om kliniese prestasie te verbeter as medies van aard. Alternatiewelik ondersoek hierdie studie die moontlike rol wat statistiese leër teorie, oftewel, statistiese algoritmes kan speel in die taak om kliniese effektiwiteit en prestasie te verbeter. Statistiese leër teorie kan beskryf word as die identifikasie van komplekse patrone in waargenome data met die oog op die voorspelling van ´n uitkoms van belang deur die onderliggende patroon wat die waargenome data teweeg gebring het na te boots en deurentyd rigiede strukturele aannames t.o.v die model struktuur te vermy.

Die vraag wat navore kom is of hertoelatings, tesame met die faktore wat ´n noemenswaardige bydrae lewer tot die manifestasie van ´n hertoelating, geïdentifiseer kan word. Indien wel, sal dit kliniese werkers kan bystaan in die taak om hertoelatings te verhoed en sodadig die kliniese prestasie van hospitale te verbeter. Die oomblik wat die statistiese model die pasiënt as ´n risiko geval identifiseer, sal dit kliniese werkers die geleentheid gee om vroegtydig op te tree om sodoende die voorkoming van ´n hertoelating te bewerkstellig. Asook, die statistiese model sal waarskynlikhede verskaf wat gebruik kan word om die hertoelatingskoers van hospitale aan te pas vir die graad van risiko wat ervaar is.

Die data wat beskou word in hierdie studie is pasiënt data wat ingesleutel word gedurende ´n besoek aan ´n hospitaal. Die privaat gesondheidsorg maatskappy betrokke is Mediclinic Internasionaal. Die betrokke data word gegenereer in die Suidelike Afrika platform van Mediclinic Internasionaal.

Daar bestaan verskeie statistiese algoritmes en modelle wat die uitkoms van hertoelatings kan modelleer. Sommige tegnieke is goed bekend, byvoorbeeld besluitnemingsbome, terwyl ander tegnieke soos neurale netwerke minder alledaags is. Logistiese regressie is nog ´n voorbeeld van ´n bekende tegniek. Ondersteunings vektor masjiene is minder bekend en ook meer kompleks. *SAS Enterprise Guide* is die gekose sagteware vir die data voorbereiding in hierdie studie, terwyl

SAS Enterprise Miner sagteware is wat gebruik word vir die modellering. Die oogmerk van hierdie studie is, eerstens, om lig te werp op statistiese leër terorie tesame met die statistiese tegnieke wat daarmee gepaard gaan. Tweedens is die studie ten doel om statistiese modellering te gebruik om hertoelatings met bevredigende akkuraatheid te voorspel.

Table of contents

List of tables

List of appendices

List of abbreviations and/or acronyms

ATC	Anatomical therapeutic chemical
AUC	area under the receiver operator curve
BMI	Body mass index
CPT	Current procedural terminology
DRG	Diagnosis related group
EHR	Electronic health records
ETL	Extract transform load
GLM	Generalised linear model
ICD – 10	10th revision of the International Statistical Classification of Diseases and Related Health Problems
HAC	Hospital acquired infection
LOS	Length of stay
RF	Random forest
SVM	Support vector machine
ROC	Receiver operator characteristic
MCSA	Mediclinic Southern Africa
MCR	Misclassification rate
MSE	Mean squared error
T	time
PPV	positive predicted value

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Clinical performance and cost efficiency are key focus areas in the healthcare industry since providing quality and affordable healthcare is a continuing challenge. The cost associated with healthcare consists of, amongst others, the cost of pharmacy products, usage of equipment and remuneration of staff. Possible causes of the high cost in developing countries, such as South Africa, are currency differences between the developing countries and the countries in which the leading pharmaceutical companies, medical equipment manufacturers and developers of the latest medical technologies are located. Additionally, maintaining facilities, retaining qualified, as well as sought after staff, adds to the financial burden experienced by healthcare providers. As a result, there is increased pressure on healthcare providers to improve efficiency and consequently, reduce cost while, upholding exceptional clinical performance. Hence, efficiency in healthcare can be described as the reduction of the cost per patient (whether for one visit or across several visits), without it being to the detriment of the clinical outcome of the admission to hospital.

Only a limited number of South Africans have access to private healthcare. The reason being the inability to afford private healthcare by means of cash payments, or the absence of medical aid cover. Both the former and the latter can be the aftermath of an unrelenting economic climate, which leads to a rise in the cost of living and unemployment. Consequently, it increases pressure on healthcare providers to attract the largest portion of this, possibly decreasing in size, subset of people. In order to attract large volumes of patients, the healthcare provider must be a provider of choice to the public, as well as, to doctors and surgeons, but also ensure that it remains on the networks of medical aids. Healthcare providers can achieve this by upholding clinical performance while implementing measures to reduce cost. Medical aids want assurance that the healthcare provider is implementing strategies to improve efficiency.

The goal of this research is to use statistical analyses and machine learning to improve efficiency in healthcare by principally considering readmissions. Patients readmitted to hospital can indicate poor clinical care and have immense cost implications. It is advantageous if readmissions can be limited.

A possible definition for a hospital readmission is (Yu, Farooq, van Esbroeck, Fung, Anand & Krishnapuram, 2015: 90):

…an admission to a hospital within a certain time frame (which can be 7, 15, 30, 60, 90 days or even as long as one year), following an original (index) admission and discharge.

Generally, stakeholders regard strategies to address the clinical performance of healthcare providers, for instance readmission rate, as mainly clinical in nature. However, this study will investigate the role of statistical modelling and machine learning in addressing the problem of readmissions. This study defines machine learning as the identification of complex patterns (linear or non – linear) present in observed data, with the goal of predicting a certain outcome for new cases by mimicking the true underlying pattern in the population which led to the observed outcomes in the sample while throughout limiting rigid structural assumptions.

The question at hand is whether patients who are at risk of readmission can be identified, along with the risk factors that can be associated with an increase in the likelihood of the event of readmission occurring. If yes, this can provide an opportunity to reduce the number of readmissions and thus avoid the resulting cost and clinical consequences. Once identified as a patient at risk for readmission during hospitalisation, it will provide an opportunity for clinical intervention. In addition, a readmission model will provide the opportunity to calculate risk scores for patients, which in turn will enable risk adjustment of the readmissions rates reported to stakeholders. A facility or doctor admitting a great deal of high − risk patients can expect to have more patients experiencing the event under consideration (readmission). Healthcare providers can consider this when it comes to performance evaluation of facilities and doctors in terms of readmission rate.

Neumann, Holstein, Le Gall and Lepage (2004: 98) affirm that risk adjusting the readmission rates of facilities by utilising a readmissions model is necessary. Neumann *et al.* (2004: 98) explain this by mentioning that risk adjustment of readmission rates will ensure that the case – mix differences of hospitals and facilities are considered when reporting on and comparing an indicator such as readmission rate. Additionally, the utilisation of the readmission model in reporting readmission rates will enable healthcare providers to eliminate the effect that factors outside of the control of the hospitals have on the readmission rate (Neumann *et al.*, 2004: 98).

The data under consideration in this study is healthcare data generated by the operations of international healthcare provider, Mediclinic International. The data under consideration is patient data captured at a hospital level in all Mediclinic hospitals, operational in Mediclinic International's Southern African platform. The data includes information from patients that is captured on admission. Additionally, data describing the nature of the patient's visit to hospital is accessible. This includes several financial and clinical measures and indicators.

Several statistical algorithms exist to model the response of interest. The techniques consist of simple, well – known techniques as well as techniques that are more advanced. Logistic

regression and decision trees are examples of simple techniques, while neural networks and support vector machines are more complex. Generally, techniques that are more intuitive are preferred in practice, not only due to the ease in explaining the technique to stakeholders, but also, stakeholders tend to trust the results of logical models, such as decision trees, more than abstract models like neural networks.

A training dataset is necessary to train (in other words estimate) a possible function, also referred to as a model, which best captures the observed patterns in the data. The goal is to obtain an estimated model that has generalisation capabilities. That is, the ability to predict the outcome of interest, being either quantitative (regression) or categorical (classification), on data not involved in the training of the model. As mentioned, there exist several algorithms to assist in obtaining such a model.

As part of the research significant explanatory variables describing the outcomes of interest will be identified, based on statistical modelling and variable selection techniques. In order to provide sufficient resources to the algorithms to detect patterns in the data, undetectable to the human eye, several variables will be provided as input to the algorithms. The variables provided as input will consist of clinical indicators and other measures or factors that potentially can predict the outcome of readmission.

Data preparation forms a substantial part of this study due to the growing size of data available and consequently the numerous possible variables that is available to provide as input to the model. Hence, Duggal, Shukla, S., Chandra, Shukla, B. and Khatri (2016a: 469) mention the application of feature selection prior to the training of a classification algorithm.

The data reside in different databases and several queries are necessary to extract the data. The data preparation includes the construction of certain variables of interest, not explicitly available in the raw data. An example of this could be the construction of a variable to indicate the number of times a patient visits the theatre during a hospitalisation. Transaction detail, for example the number of theatre invoice numbers on the account, can provide this information.

The outcome variable of interest in this study, is an indicator variable indicating whether the patient admitted, is a hospital readmission. That is, the patient is readmitted within 30 days after being discharged. Generally studies refer to the initial admission as the index admission and to the subsequent admission, given occurrence within 30 days after discharge, as the readmission (Yu *et al.*, 2015: 93).

Certain logic is necessary to identify the index admission and readmission. The nature and complexity of this logic is determined largely by the admission and billing protocol of a hospital. Certain healthcare providers allocate one account number per patient, thus every time the particular patient is admitted to hospital the same account number, but possibly different visit

keys, is allocated to the patient. Thus, the index admission and readmission within 30 days can easily be determined and flagged. Another approach is to allocate a new unique account number on every occasion a particular patient visits the hospital. This approach makes the identification of readmissions more difficult. Since it results in dependence on information like country issued identification number, passport number, name and surname to link the unique account numbers of one patient across several visits, in order to determine if the subsequent admission to the index admission is indeed a readmission within 30 days.

Even though the data used in this research are anonymised and do not include patient names, nor patient addresses, the data is treated as highly confidential. Explicit reporting of clinical performance, for example readmission rate, of a hospital or Mediclinic South Africa as a whole, does not occur in this study. The study focusses mainly on methodology and the machine learning techniques under consideration as opposed to clinical performance comparison per se.

Two applications that exist once a readmission model is available is, implementing the model into the data warehouse to automatically calculated a readmission risk score for every patient as the patient record comes from hospital level into the data warehouse (Billings, Blunt, Steventon, Georghiou, Lewis & Bardsley, 2012: 2). Alternatively, personnel in the hospitals can enter the patient's input features (characteristics) applicable to the readmission model, as obtained from the patient files, onto a device and a risk score will be available at discharge (Billings *et al.*, 2012: 2).

Possible factors that lead to the occurrence of hospital readmissions are: quality of care during hospitalisation, social circumstances of the patients, the nature of the condition of the patients and the patients' general state of health (Zheng, Zhang, Yoon, Lam, Khasawneh & Poranki, 2015: 7111). Shams, Ajorlou and Yang (2015: 19) mention that the Medicare Payment Advisory Commission points out that hospital acquired complications (HAC), complications in general, inadequate follow up arrangements, poor discharge procedure and the lack of medication reconciliation can all lead to an increase in the likelihood of a patient being readmitted.

It is of utmost importance to gain familiarity with the exact causes of readmissions and consequently enable hospital personnel to implement specific interventions on high − risk patients to reduce the risk of being readmitted (Zheng *et al.*, 2015: 7111).

1.2 PROBLEM STATEMENT

A collective viewpoint of studies concerning readmissions is that a certain percentage of readmissions can be avoided should a patient be identified as a high – risk case, either before admission, during hospitalisation or after discharge. Also, most studies are in agreement with the stance that should healthcare providers be able to avoid a fraction of the readmission occurring

at its facilities, substantial cost saving is possible. However, it remains a challenge to pinpoint patients that are considered as high − risk cases for readmission within say 30 days after being discharged (Zheng *et al*., 2015: 7111). A reason for this, as Zheng *et al*. (2015: 7111) point out, is the intricate nature of the factors that lead to the readmission of a patient. McIlvennan, Eapen and Allen (2015: 1796) affirm this by referring to hospital readmissions as “multifactorial”.

In order to advance efficiency, interventions are limited predominately to high – risk patients due to the cost associated with these interventions (Futoma, Morris & Lucas, 2015: 229). It is of no use if the cost of interventions to reduce readmissions overshadow the reduction in cost by successfully preventing readmissions. In addition, Yu *et al.* (2015: 90) point out that generally hospitals will have to implement interventions at their own cost, without reimbursement from insurance companies/medical aids. Billings *et al.* (2012: 7) suggests that different interventions can be applied to different risk intervals, typically the more expensive intervention can be implemented for the high − risk patients and the less expensive interventions for the lower − risk patients. Some consensus exists in this regard since other studies also share the idea of different levels of interventions based on the magnitude of the readmission risk for a patient with the focus of enhancing the cost efficiency of the preventative measures (Shadmi, Flaks - Manov, Hoshen, Goldman, Bitterman & Balicer, 2015: 283; Tong, Erdmann, Daldalian, Li & Esposito, 2016: 2).

Internationally instances exist where healthcare providers are penalised in a particular manner for a high number of readmissions (Zheng *et al*., 2015: 7110). Readmission rate is a key factor on which healthcare providers report as part of annual reporting of clinical outcomes. Implementing measures on patients identified as high − risk cases for readmission can be advantageous in negotiations with medical aids to remain on the medical aid’s network as a healthcare provider of choice.

1.3 LITERATURE REVIEW

In the literature, several studies identify the necessity of building models to predict clinical outcomes such as readmissions. The literature describes, not only the reason for the construction of such models, but also the data sources and methodology involved. Descriptions of the data source includes information on the inputs under consideration, as well as, the data preparation process. In contrast, explanations of the methodology refer to the various machine learning algorithms implemented. In addition, detailed discussions on the results, which include the listing of significant variables, as well as the respective performance of the models, are available. This chapter briefly highlights some aspects regarding related research available in the literature.

Once a facility has the ability to identify the patients that are of high − risk to be readmitted, the literature proposes intervention strategies such as follow − up visits and contacting patients by means of phone calls after discharge (McIlvennan *et al.,* 2015: 1797). Futoma *et al*. (2015: 229)

indicate that the enhancement of patient knowledge and understanding of the episode of clinical care received, as well as provisioning of prognosis after discharge to the patient's primary doctor can contribute in preventing readmissions. Informing patients about the advantages that prevention of readmission can pose and how it can be achieved is also a proposed intervention (Tong *et al.*, 2016: 2). The increase popularity and access to various forms of communication for example emails and other communication applications via the internet can be of great assistance. The availability of a toll free helpline for patients to clarify uncertainties regarding after care or medication use, is another example of an intervention (Billings *et al.*, 2012: 8).

McIlvennan *et al.* (2015: 1799) mentions monitoring mortality and length of stay (LOS) while attempting to reduce the number of readmissions. The reason for this can be the fact that deceased patients cannot be readmitted, thus if the number of deaths increase, the number of potential readmissions will decrease. It is for this reason that it is sensible to calculate the readmission rate for a particular period as the ratio of the number of readmissions and the number of discharges, rather than having the number of admissions as the denominator. It is worth noting that, increasing LOS thoughtlessly to reduce readmissions will nullify the financial advantages associated with the possible reduced number of readmissions.

Although modelling of readmissions is common there are few readmission models that are constructed on the data of the South African healthcare environment. Predictive models calibrated in a particular healthcare environment, for example a specific country, are most likely not able to deliver the same predictive performance on healthcare data of another country (Billings *et al.*, 2012: 2). Although universal readmissions models exist, for example the LACE model, inferior performance is common and the diverse characteristics of different patient populations among hospitals are a likely cause of the inferior performance (Yu *et al*., 2015: 89). The name of the LACE score calculation arises from the measures involved in the calculation namely, LOS (L), acuity level of the condition treated (A), presence of comorbidities (C) and utilisation of the emergency center (E) (Zheng *et al.*, 2015: 7110).

Yu *et al.* (2015: 89) reports improved performance of models that predict readmission risk at discharge compared to models that predict readmission risk at admission. However, Yu *et al.* (2015: 94) mentions an interesting result that the predictive machine learning models using data present at admission, frequently outperformed or matched the performance of the LACE model, which is used only once the patient has been discharged. This study will consider predictions at discharge. The reason being, as Walsh and Hripcsak (2014: 420) identify, the clinical coding of patients for the current visit to hospital is not available at admission but only at discharge. Although the availability of readmission risk at admission will provide clinical workers with more time to implement interventions, this study believes that the approach to apply interventions once a model identifies a patient as a risk for readmission at discharge, can be effective.

Duggal *et al*. (2016a: 470) point out that missing values and unbalanced response variables generally contaminate healthcare data. Healthcare data also tend to be of high dimension and a proposed remedy for this phenomenon is feature selection by means of correlation analysis or chi – squared calculations (Duggal *et al*. 2016a: 473).

Duggal *et al*. (2016b: 521) neither consider admissions younger than 18 years, nor maternity related patients. This study's reason for the exclusion of maternity patients is discussed in Section 2.3.2. Walsh and Hripcsak (2014: 419) also omitted patients younger than 18 years as well as patients admitted for "normal delivery" or rehabilitation. Most studies also exclude mortalities. Tong *et al.* (2016: 3) exclude patients with an index admission for psychiatry, rehabilitation, maternity and also exclude mortalities and all new born admissions. The possible impact of socioeconomic features on readmission of patients obtain consideration (Jiang *et al*., 2003 cited in Duggal *et al.*, 2016b: 520). This study will not be able to investigate socioeconomic variables as possible predictors of readmissions due to unavailability of the appropriate data. Yu *et al.* (2015: 90) identify this as a possible limitation in the quest of successfully modelling readmissions.

Yu *et al.* (2015: 91) mention the use of levels of severity of prognosis in predicting readmission risk in certain studies. Shams *et al*. (2015: 20) agree that the severity of the patient's condition can be explanatory of readmission probability. The severity of the patient's condition is partially encapsulated in the comorbidities and complications present, which this study obtains from the clinical coding on the account.

It is noticeable that presenting the algorithms with appropriate inputs can distinguish a model form the rest. This may involve manipulation of raw data in order to obtain variables that may be useful (Duggal *et al.*, 2016b: 522). The number of historic admissions, number of visits to specialists and the type of specialists seen in the run to the current admission can be included as features (Billings *et al.*, 2012: 3).

Walsh and Hripcsak (2014: 419) uphold that often the attempts to predict readmissions vary between two approaches, namely, disease specific prediction of readmission or prediction of readmission risk in general. The decision of whether a disease specific or non − disease specific readmission model is under consideration is decisive and in case of the former, specification on what disease to consider is influential (Walsh & Hripcsak 2014: 419).

This study considers modelling readmission simultaneously by means of one model across all clinical groups. Clinical groups refer to a grouping of conditions or procedures that is related. This approach obtains support by the observation that the reason for readmissions often differs or is unrelated to the reason of the initial admission (McIlvennan *et al*., 2015: 1799). However, the clinical grouping of the index admission as an input variable to the model can be advantageous. Duggal *et al*. (2016b: 520) state that modelling readmissions by focussing on a particular disease

is advantageous in the sense that patients have disease specific characteristics that influence their probability of readmission. Walsh and Hripcsak (2014: 425) agree and mentions that certain factors, for example blood test results, can be a significant predictor for some diseases more than for others.

Futoma *et al*. (2015: 232) investigate both the modelling across all DRG's (Diagnosis related groups), as well as modelling each DRG separately. Futoma *et al.* (2015: 232) continues by comparing the performance of the overall model versus the DRG − specific model in each of the DRG's. Futoma *et al.* (2015: 233) also report on the presence of a noteworthy correlation between the best AUC (area under the receiver operator curve) and the readmission rate per DRG, as well as between the greatest AUC and the number of readmissions within the DRGs. On the contrary, Futoma *et al.* (2015: 233) state that a weak correlation exists between the best AUC and the number of admissions per DRG. Futoma *et al.* (2015: 233) explain that the presence of the former correlations (between AUC and readmission rate/number of readmissions) indicates that DRGs where readmissions are common are easier to model accurately. On the other hand, DRGs with more admissions are not necessarily easier to model more accurately (Futoma *et al.*, 2015: 233). This aligns with the discussion of class imbalance in the response variable. Chapter 4 contains further details regarding class imbalance and remedies to circumvent the phenomenon.

Typically, studies consider 30 − day readmissions (i.e. 30 days between discharge date of index admission and admission date of readmission), but there is debate regarding which period is optimal. An important time component exits in readmissions since it is possible that the occurrence of readmissions shortly after the initial hospitalisation's discharge, is likely due to the quality of care during the initial hospitalisation (Shams *et al*., 2015: 23). While, readmissions separated by a longer period from the discharge of the initial hospitalisation, are likely due to insufficient care after discharge (Shams *et al.*, 2015: 23). The insufficient care can include lack of follow – up visits, lack of clinical knowledge by the patient or caregiver of the patient at home or the complete lack of assistance for the patient at home.

At least one study incorporates a different approach by considering a Cox regression (Yu *et al.*, 2015: 95). The advantage of this approach is the time interval chosen, for example 30 – days, has less of an influence on the results since the outcome is not as rigid (1/0) as the case is in the classification machine learning techniques (Yu *et al.*, 2015: 95). For example, the difference between a readmission after 30 days and 32 days is smaller when using a Cox regression, as opposed to the machine learning classification techniques (Yu *et al.*, 2015: 95). Using machine learning, the former will be allocated to the readmission group whereas the latter will not be allocated to the readmission group, even though the difference is only two days.

Considering the possible techniques to use, Zheng *et al.* (2015: 7111) endeavors to use random forests (RF), support vector machines (SVM) and neural networks to predict readmissions.

Alternatively, Duggal *et al.* (2016b: 522) considers naïve Bayes, logistic regression, random forests, Adaboost and neural networks in an attempt to model readmissions for diabetic patients due to the superior ability these techniques possess to model a binary response variable. Tong *et al.* (2016: 2) prefer models that provide a straightforward explanation of results over the typical machine learning techniques such as random forests. Possibly the most popular readmission modelling technique in the literature is multivariate logistic regression (Duggal *et al.*, 2016b: 520).

1.4 CHAPTER OUTLINE

Chapter 1 serves as an introduction to readmission modelling and briefly highlights what similar work in the literature encompass. Chapter 2 describes the setting this study focusses on, especially in terms of the data used. Chapter 2.2 continues by providing background on Mediclinic International. Chapter 2.3 mainly focusses on the variables of interest, together with details on the construction of the variables. Initial feature engineering is described in Chapter 2.3. Feature engineering and manipulation of variables are also presented in Chapter 4. Chapter 2.4 briefly provides insight to some distributional characteristics of healthcare data.

Chapter 3 summarises the four machine learning techniques relevant to this study. This chapter contain explanations of the algorithms, mathematical derivations and discusses the advantages and disadvantages with respect to the techniques. Chapter 3 also focusses on explaining SAS Enterprise miner's functionality with respect to the techniques under consideration. Chapter 4 provides an in − depth discussion of the modelling process by making use of the data described in Chapter 2 and the modelling techniques described in Chapter 3. Comparison between the different models in terms of fit statistics is presented in this chapter. Finally, Chapter 5 provides a summary of the findings of this study. The chapter identifies accomplishments, shortcomings and areas for further research.

CHAPTER 2
DESCRIPTION OF DATA

2.1 INTRODUCTION

This chapter aims to provide a description of the data that this study utilises. Background on the owner of the data, Mediclinic International, forms part of the discussion. A discussion regarding the variables of interest, as well as the variables constructed form the raw data follows. The chapter emphasises that given the possession of an abundance of data, the data tend to reside in different databases or tables and usually does not comprise of a single dataset that is equipped to train a model on. In addition, the variables of interest are not customarily all explicitly available in the data but require some data manipulation in order to include the variables in a model building process.

The process of data preparation utilises statistical techniques, for example decision trees, to provide assistance in the manipulation of the raw data.

2.2 BACKGROUND ON MEDICLINIC INTERNATIONAL

The private hospital group, Mediclinic International, is operational in three platforms, namely, Southern Africa, Switzerland and the Middle East. The South African platform consists of hospitals in South Africa and Namibia. According to the official website of Mediclinic International, the total number of hospitals operational in the Southern African platform is 51 with three additional day clinics. Mediclinic International claims to have more than 8000 beds operational in the Southern African platform.

Mediclinic International further states that the Switzerland platform consists of 17 hospitals and 4 outpatient facilities with more than 1800 inpatient beds in total. Mediclinic Middle East consists of 29 facilities (7 hospital and 22 clinics) with more than 900 inpatient beds being operational in the United Arab Emirates. Finally, Mediclinic International owns 29.9 percent of the Spire Healthcare group.

Regarding stock exchange listings, Mediclinic International is listed on the London Stock Exchange, as well as a secondary listing on the Johannesburg Stock Exchange. Mediclinic International is also listed on the Namibian Stock Exchange. Today, more than 30 years after the company which is now known as Mediclinic International started, it is one of the largest private hospital groups in the world.

2.3 VARIABLES CONSIDERED

The data that the study focusses on are patient data captured at a hospital level. This includes several patient characteristics, duration and date of the hospital stay, as well as, financial information of each visit to hospital. The data of interest comprises only of patients of the Southern African platform of Mediclinic International. Mediclinic International's Southern African platform do not use an EHR (electronic health record) system. Consequently, the medical history of patients, for example chronic medication and comorbidities is not available. Therefore, this study predominately relies on coding info (ICD – 10 and CPT codes) of previous visits to hospital to obtain an indication of the patients' status of health. This is a noteworthy disadvantage compared to several other studies in the literature and may be to the detriment of the modelling performance reported in this study. Briefly stated, ICD – 10 (International statistical classification of diseases) codes refer to codes used to describe medical (non – surgically related) conditions for example Pneumonia. While, CPT (current procedural terminology) codes describes surgically related procedures, such as a caesarean section.

After collection of data at hospital level, staging of the data in a data warehouse by means of ETL (extract, transform, load) processes occur. Mediclinic International has dedicated data warehouse specialists to perform this task. The analytics department of Mediclinic International predominantly works with SAS products, for example SAS Enterprise Guide. Consequently, the construction of SAS datasets from the data staged in the data warehouse follows. Finally, the data resides in several SAS datasets and the preparation of a base table from which modelling is performed, comprises of several data queries, as well as, joins of the respective SAS datasets. Joining two datasets includes using an unique identifier present on both datasets to create a new dataset consisting of information from both initial datasets. In addition, construction of variables not explicitly available is necessary. The subsequent sections of Chapter 2 describe the data elements of interest and shed light on the construction of the variables where necessary.

2.3.1 Number of observations

The data of interest consists of all hospital admissions (in – patients) for several consecutive years. A hospital admission is a patient that is allocated a bed and thus has accommodation days billed. This does not necessarily imply that the patient overnighted in the hospital, as a patient can be allocated a bed for the duration of a single day. The dataset under consideration comprises of 1 798 802 observations, which decreases to 1 705 064 due to certain exclusions. The subsequent sections of Chapter 2 discuss these exclusions.

The fact that the data do not involve financial variables, especially regarding the response variable, is advantageous due to the inflation present in financial measures across different years.

Tariff negotiations occurring in the beginning of a calendar year, as well as yearly pharmacy inflation have a significant effect on financial measures across calendar years.

2.3.2 Response variable

The length of time between a consecutive discharge and admission of the same patient for it to be classified as a readmission, is critically debated (Vaduganathan *et al*. cited in McIlvennan *et al*., 2015: 1800). As mentioned before, readmissions shortly after discharge may be linked to the quality of the episode of care in hospital, while readmission after 30 days may be a result of the advanced severity of the index admission or may be due to factors unrelated to the quality of care provided during the initial admission (McIlvennan *et al.,* 2015: 1800).

If a patient is admitted several times within 30 days, only the first visit after the index (initial) admission acts as a readmission for that index admission (Yu *et al.*, 2015: 93). It is similar to the approach of Billings *et al.* (2012: 3) where only the admission occurring within 30 – days but immediately after the previous admission forms an admission/readmission pair. With regards to this study's approach, given the occurrence of three admissions of the same patient within a period of 30 – days, the first and second admission will form an admission/readmission pair and the second and third admission will form an admission/readmission pair, but the first and third admission do not form an admission/readmission pair even though the time difference between the discharge date of the first admission and admission date of the third admission is less than 30 – days. Thus, a readmission for a particular index admission can in turn also be the index admission of a subsequent readmission.

Tong *et al.* (2016: 3) warns that the situation where one patient contributes to several index and readmission pairs, can lead to the presence of correlation between the observations in the dataset. However, the proposed solutions to counter this phenomena do not result in an improvement in the model performance, thus the possible correlation is ignored (Tong *et al.*, 2016: 3).

With regards to this study, the response variable is an indicator variable (binary) of whether the respective admissions to hospital are a readmission, with a time restriction of less than or equal to 30 days separated from an index admission. In the context of this study, a readmission is an admission to hospital within 30 days after the discharge date of the most recent visit to hospital of a particular patient. This study also considers a readmission, after any number of days, as an index admission for a possible subsequent admission that will act as the new readmission. Thus, a readmission corresponding to a prior index admission can also act as the index admission of a subsequent admission (readmission).

The rest of this section describes the methodology in identifying a patient as a readmission. Each visit to hospital triggers the assignment of a new account number to the patient that in turn, based

on a particular algorithm, gives rise to a unique key. Thus, the different unique keys for the same patient are matched together in order to identify an admission as either a 30 – day readmission, or not, based on the time elapsed between the consecutive admissions and given that the same patient is involved in both admissions.

According to *SAS Institute Inc*. (2018a), SAS Enterprise Guide has the ability to construct *datetime* values and SAS stores the *datetime* value, which is constructed from a date field and a time field, as the number of seconds that has passed since 1 January 1960. The discharge date and discharge time of the index admission is combined to form a discharge *datetime* value. In addition, for the subsequent admission for each patient, the admission date and admission time is also converted to an admission *datetime* value. The difference (converted to the number of days) between the two *datetime* values determines the value of the response variable. If the difference is smaller than or equal to 30 then the admission is a readmission (indicated by the number 1), otherwise it is not a readmission (indicated by the number 0).

As mentioned, each admission can act as an index admission, thus an admission classified as a readmission, can in turn act as the index admission for a subsequent admission. This study considers all – cause readmissions and only a limited number of exclusions occur. Shams *et al*. (2015: 19) put emphasis on the fact that in the literature little studies exist that distinguish between planned and unplanned readmissions. Descriptions of the exclusions follow in the subsequent sections. It is noteworthy that the dataset is imbalanced since the proportion of events (30 – day readmission) is rare compared to the non – events. A discussion on the techniques to address this imbalance in the dataset, follows in Chapter 4.

2.3.2.1 Discharge type of the index admission

Different discharge types may occur in the data of Mediclinic Southern Africa (MCSA). Certain discharge types may indicate poor care, more than other discharge types. Any admission occurring within 30 days after an initial admission with one of the following discharge types, is not classified as a readmission and is removed from the dataset:

I. Split or Partial Account (if an account is split, then one admission is represented by two accounts (thus also two keys) and will appear as an index/readmission pair with the readmission occurring on the same date as the index admission's discharge date).
II. Weekend discharge – will be returning
III. Deceased
IV. Operation deferred – patient related reason
V. Operation deferred – hospital related reason
VI. Patient removed by state authorities
VII. Patient institutionalised – mental etc.

The reasoning for excluding the preceding observations is the fact that these admissions were either not readmitted due to the discharge type (e.g. deceased), or the patients was readmitted due to the discharge type (e.g. Weekend discharge – will be returning).

2.3.2.2 Maternity cases

All admissions occurring within 30 days of an index admission with the second admission classified as an elective maternity admission is removed from the dataset, since an elective maternity admission is unavoidable and is due to happen. The quality of care during the index admission cannot prevent the subsequent admission in this instance.

2.3.3 Continuous explanatory variables

Several continuous variables form part of the dataset. Alternatively, these variables can be included as categorical variables (see Section 2.3.4, as well as Chapter 4). The variables are (corresponding to the index admission):

I. Age at admission
II. Theatre minutes
III. Accommodation days
IV. ICU days
V. High Care days
VI. Prosthesis amount billed
VII. BMI (body mass index)
VIII. Number of theatre events during one period of clinical care at a hospital

Patients with a patient type of “maternity” obtain a missing value for BMI since these patients will have a high BMI. It is possible to break down theatre minutes into major theatre minutes charged and minor theatre minutes charged. Typically, billing of a more complex procedure occurs by means of major theatre minutes which has a higher associated tariff. Indicator variables of whether major theatre minutes are present in the billing of the account are of interest and can indicate the complexity of the procedure.

The number of theatre events are the number of theatre invoice numbers on an account. However, due to billing practices and clinical processes multiple theatre invoice numbers can be misleading in indicating multiple theatre events. For example, in certain cases, a patient can undergo two procedures. Billing of the two procedures occurs separately. The theatre out − time allocated to the invoice number of the first procedure (although the patient never left the theatre) will be close to the theatre in − time allocated to the invoice number of the second procedure. The construction process of the dataset avoids this phenomenon by not counting theatre events with

an initial time – out and a consecutive time – in difference of less than 60 minutes, as multiple theatre events.

2.3.4 Categorical explanatory variables

Categorical explanatory variables also form part of the dataset (corresponding to the index admission):

I. Gender.
II. Arrival method (via ambulance, walk – in, via emergency room, Transfer from other facility, via helicopter, born).
III. Major theatre indicator (binary) i.e. major theatre minutes billed versus major theatre minutes not billed.
IV. ICU indicator (binary) i.e. patient was admitted to the ICU versus the patient was not admitted to the ICU.
V. High Care indicator (binary) i.e. patient was admitted to the high care unit versus the patient was not admitted to the high care unit.
VI. Prosthesis indicator (binary) i.e. patient obtained a form of prosthesis versus no prosthesis obtained.
VII. Cathlab minutes billed indicator (binary) i.e. the patient underwent a procedure in the cathlab (specialised theatre) during the index admission or not.
VIII. Admission over weekend indicator (binary)
IX. Discharge over weekend indicator (binary)
X. Season admitted (Winter/Autumn versus Summer/Spring)
XI. Admission time (early morning, morning, afternoon, evening)
XII. Discharge time (early morning, morning, afternoon, evening)
XIII. BMI as a categorical variable.
XIV. Pharmacy products billed indicator (see Section 2.3.4.1).
XV. Clinical grouping (final clinical category, final clinical group, final clinical sub−group).
XVI. Patient type (in − patient, day case etc.).

2.3.4.1 Pharmacy product usage indicator variables

The pharmacy products that a patient uses both in hospital and after discharge can have an impact on the likelihood of patient readmission. The following scenarios and the associated products are of interest:

I. Pain medication – often patients prescribed pain medication at discharge do not experience that the pain medication provides significant relief of the pain which can lead to readmission to hospital. The relevant ATC (Anatomical therapeutic chemical) classifications are N02 and M01.

II. Nausea medication – patients receiving primary or secondary treatment for being nauseous can often experience readmission due to nausea or vomiting that do not clear up. In addition, patients that is treated for nausea or vomiting and dehydration are likely to be readmitted. The relevant ATC classification is A03.
III. Constipation is often a cause of readmission. Should the medication not have the desired effect, the patient is readmitted to hospital for an enema. The ATC classification of interest is A06.
IV. Patients with a risk of blood clots will receive a prescription to reduce the likelihood of it occurring after discharge. However, in some instances, blood clots do occur and readmission to hospital is unavoidable. The corresponding ATC classification is B01.
V. After discharge, infections might occur. If this is a likely event to occur based on the reason for the index admission, the patient will receive treatment in terms of medication to use after discharge to prevent infections. The ATC codes J01 and D01 is applicable in this case.
VI. Patients prescribed sedatives are at risk of readmission for potential falling that might occur. The relevant ATC code is N05.

The pharmacy products belonging to the ATC groupings described above can all potentially be associated with a higher risk of readmission due to the underlying reason for it being prescribed or due to the effect the medication has on the patient. The scenarios described above, as well as, the ATC classifications were obtained based on communication with clinical stakeholders of Mediclinic International, as well as information from the webpage of WHO Collaborating Centre for Drug Statistics Methodology (2018).

The variables describing the pharmacy product usage in the dataset are binary indicator variables. The variables indicate whether any of the ATC classifications described above is present in the list of billed drugs occurring on the patients' pharmacy history of the index admission. Should a product residing in any of the ATC classifications mentioned be present on the account of a patient, the appropriate ATC Level 5 classification code is triggered and appear as binary explanatory variables. The restriction that only products billed more than 100 times across all patients under consideration is applied, in order to limit the number of indicator variables created. Chapter 4 also describes further application of feature engineering to the pharmacy product indicator variables.

2.3.4.2 Comorbidity and complications indicator variables

The presence of comorbidities and complications during an admission to hospital can potentially increase the risk of readmission. There exists plenty of potential comorbidities and complications that can be associated with the hospital admissions in the dataset. Complications and

comorbidities appear as type 2, 3 or 7 codes in the coding history of a patient. Considering all the hospital admissions during the period under consideration, there are 14 470 unique codes coded as type 2, 3 or 7 codes. Consequently, construction of indicator (binary) variables for all these codes will result in 14 470 indicator variables. An excessive number of explanatory variables increases the total training time of the models. In addition, the elimination of excessive explanatory variables are likely to prevent overfitting, as well as, positively affect the performance of the models (Christie, Georges, Thompson & Wells, 2015: 9-7). Therefore, this study conducts a process of limiting the number of codes that can act as comorbidities and complications by means of statistical feature selection that occurs separately from the training of the readmission model. The assumption that complications and comorbidities will increase the time spent in hospital per visit is a crucial component in the process to follow.

Considering only accounts with zero theatre minutes billed is an attempt to include only the medical cases. Furthermore, only accounts with a final allocated code being an ICD code is used in the process to construct a complication and comorbidity list. The Health Information Management department of Mediclinic International maintains a list of codes to be ignored, which are also excluded as possible comorbidities and complications. The ignore codes are codes that are coded in the type 2,3 or 7 positions in combination with other codes, but of their own accord do not add any information of interest. All codes occurring less than 10 times are eliminated from the dataset. Finally, only ICD codes (diagnostic codes) can act as comorbidities and complications therefore the removal of all CPT (procedural) codes.

After the process of filtering out codes as described, the number of type 2, 3 or 7 ICD codes that can potentially be published on the list of influential comorbidities and complications decreases from 14 470 to 3 274. These codes are transposed (observations become variables) to form indicator variables. For example, the variable called “code A” will have a zero if the code does not occur on the respective accounts and a one if it does. In addition, for each account the calculation of the difference in days between the discharge date and admission date for the visit to hospital is crucial. Per final clinical sub – group the median of all the patients’ difference in discharge date and admission date is calculated. Every observation whose difference in the discharge date and admission date exceeds the median for the final clinical sub – group the account belongs to, is flagged as an extended length of stay case and this leads to the construction of a binary response variable.

Modelling of the response variable on all the indicator variables constructed from the codes transpires next. The modelling occurs by means of six decision trees. Each tree trains from a stratified sample and subsequently, each decision tree proposes a certain number of significant variables, which in effect proposes codes as possible influential complications and comorbidities.

Table 2.1: Properties of the stratified samples

Sampling method	Stratified sample
Sample size (n_h)	50 000
Strata	Final clinical sub − groups
Number of strata	385
Allocation	Proportional allocation

A specific code forms part of the final comorbidity and complication list only if at least one of the trees identifies the code as significant. Table 2.2 provides the codes which at least one decision tree found significant and consequently occur on the comorbidity and complication list.

Table 2.2: Comorbidity and complication list consisting of ICD − 10 codes

Code	Code Description
A09.9	GASTROENTERITIS AND COLITIS OF UNSPECIFIED ORIGIN
B33.3	RETROVIRUS INFECTIONS NOT ELSEWHERE CLASSIFIED
B95.6	STAPH AUREUS AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B96.1	KLEBSIELLA PNEUMONIAE AS CAUSE DIS CLASS OTHER CHAPS
B96.2	ESCHERICHIA COLI AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPS
B96.8	OTHER BACT AGENTS AS CAUSE OF DIS CLASS OTH CHAPS
D64.9	ANAEMIA UNSPECIFIED
E87.5	HYPERKALAEMIA
E87.6	HYPOKALAEMIA
I10	ESSENTIAL PRIMARY HYPERTENSION
J18.9	PNEUMONIA UNSPECIFIED
J90	PLEURAL EFFUSION NOT ELSEWHERE CLASSIFIED
N39.0	URINARY TRACT INFECTION SITE NOT SPECIFIED
P07.3	OTHER PRETERM INFANTS
R 41.00	DISORIENTATION UNSPECIFIED

The list of comorbidity and complication codes will assist in the construction of comorbidity and complication indicator variables on the main dataset from which the readmission modelling will take place. An expanded list is constructed from the codes in Table 2.2. The lack of consistent occurrence in the data of closely related codes to the codes in Table 2.2 can be the reason for the failure of the decision trees to find certain codes significant. The individual occurrence rate of similar codes to those in Table 2.2 fluctuates from one clinical coder to another as coding practices or coding proficiency differ. Therefore, all codes among the 3 274 codes, starting with the first

three characters of any of the codes in Table 2.2, occur on the expanded list of comorbidity and complication codes. Related to this strategy is Walsh and Hripcsak (2014: 420) process of combining ICD codes.

The expanded list in Table 2.3 is used to create indicator variables in the main dataset from which the models in Chapter 4 are built. For a code to act as a comorbidity and complication indicator variable in the dataset from which the readmission models train, the code must appear as a type 2, 3 or 7 code on an account and the code must appear in the expanded comorbidity and complication list in Table 2.3.

Table 2.3: Expanded comorbidity and complication list consisting of ICD − 10 codes

Code	Code description
A09.0	OTHER AND UNSPECIFIED GASTROENTERITIS AND COLITIS OF INFECTIOUS ORIGIN
A09.9	GASTROENTERITIS AND COLITIS OF UNSPECIFIED ORIGIN
B33.0	EPIDEMIC MYALGIA
B33.2	VIRAL CARDITIS
B33.3	RETROVIRUS INFECTIONS NOT ELSEWHERE CLASSIFIED
B33.4	HANTAVIRUS PULMONARY SYNDROME
B33.8	OTHER SPECIFIED VIRAL DISEASES
B95.0	STREPT GROUP A AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.1	STREPT GROUP B AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.2	STREPT GROUP D AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.3	STREP PNEUMONIAE AS CAUSE OF DIS CLASSIF OTHER CHAPTERS
B95.4	OTHER STREP AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.5	UNSTREP AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.6	STAPH AUREUS AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.7	OTHER STAPH AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPTERS
B95.8	UNSTAPH AS CAUSE OF DIS CLASSIF TO OTHER CHAPTERS
B96.0	MYCOPLASMA PNEUMONIAE AS CAUSE DIS CLASS OTH CHAPS
B96.1	KLEBSIELLA PNEUMONIAE AS CAUSE DIS CLASS OTHER CHAPS
B96.2	ESCHERICHIA COLI AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPS
B96.3	HAEMOPHILUS INFLUENZAE AS CAUSE OF DIS CLASS OTH CHAPS
B96.4	PROTEUS (MIRABILIS)(MORGANII)CAUSE OF DIS CLASS OTH CHAPS
B96.5	P(AERUGIN)(MALLEI)(PSEUDOMALLEI)CAUS DIS CLASS OTH CHAP
B96.6	BACILLUS FRAGILIS AS CAUSE OF DIS CLASSIFIED TO OTHER CHAPS
B96.7	CLOSTRIDIUM PERFRINGENS AS CAUSE OF DIS CLASS TO OTH CHAPS
B96.8	OTHER BACT AGENTS AS CAUSE OF DIS CLASS OTH CHAPS
D64.0	HEREDITARY SIDEROBLASTIC ANAEMIA
D64.1	SECONDARY SIDEROBLASTIC ANAEMIA DUE TO DISEASE
D64.2	SECONDARY SIDEROBLASTIC ANAEMIA DUE TO DRUGS AND TOXINS
D64.3	OTHER SIDEROBLASTIC ANAEMIAS
D64.4	CONGENITAL DYSERYTHROPOIETIC ANAEMIA
D64.8	OTHER SPECIFIED ANAEMIAS
D64.9	ANAEMIA UNSPECIFIED

E87.0	HYPEROSMOLALITY AND HYPERNATRAEMIA
E87.1	HYPO-OSMOLALITY AND HYPONATRAEMIA
E87.2	ACIDOSIS
E87.3	ALKALOSIS
E87.4	MIXED DISORDER OF ACID-BASE BALANCE
E87.5	HYPERKALAEMIA
E87.6	HYPOKALAEMIA
E87.7	FLUID OVERLOAD
E87.8	OTHER DISORDERS OF ELECTROLYTE AND FLUID BALANCE NEC
I10	ESSENTIAL PRIMARY HYPERTENSION
J18.0	BRONCHOPNEUMONIA UNSPECIFIED
J18.1	LOBAR PNEUMONIA UNSPECIFIED
J18.2	HYPOSTATIC PNEUMONIA UNSPECIFIED
J18.8	OTHER PNEUMONIA ORGANISM UNSPECIFIED
J18.9	PNEUMONIA UNSPECIFIED
J90	PLEURAL EFFUSION NOT ELSEWHERE CLASSIFIED
N39.0	URINARY TRACT INFECTION SITE NOT SPECIFIED
N39.1	PERSISTENT PROTEINURIA UNSPECIFIED
N39.2	ORTHOSTATIC PROTEINURIA UNSPECIFIED
N39.3	STRESS INCONTINENCE
N39.4	OTHER SPECIFIED URINARY INCONTINENCE
N39.8	OTHER SPECIFIED DISORDERS OF URINARY SYSTEM
N39.9	DISORDER OF URINARY SYSTEM UNSPECIFIED
P07.0	EXTREMELY LOW BIRTH WEIGHT
P07.1	OTHER LOW BIRTH WEIGHT
P07.2	EXTREME IMMATURITY
P07.3	OTHER PRETERM INFANTS
R 41.00	DISORIENTATION UNSPECIFIED
R 41.10	ANTEROGRADE AMNESIA
R 41.20	RETROGRADE AMNESIA
R 41.30	OTHER AMNESIA
R 41.80	OTH/UNSP SYMPT & SIGNS INVOLV COGNITIVE FUNCT & AWARENESS

However, if the code appears on the list and is coded as a type 2, 3 or 7 code, then the code is not considered as a comorbidity or complication if the code is similar to the final ICD code allocated to the account. The reason is that if a code is the same as the final ICD code then the code cannot act as a comorbidity or complication but rather as a primary diagnosis. Finally, if the code is a CPT code then it cannot act as a comorbidity or complication as only ICD codes are relevant. Diagram 2.1 illustrates the algorithm of the construction of the comorbidity and complication indicator variables on the main dataset from which the readmission models will train.

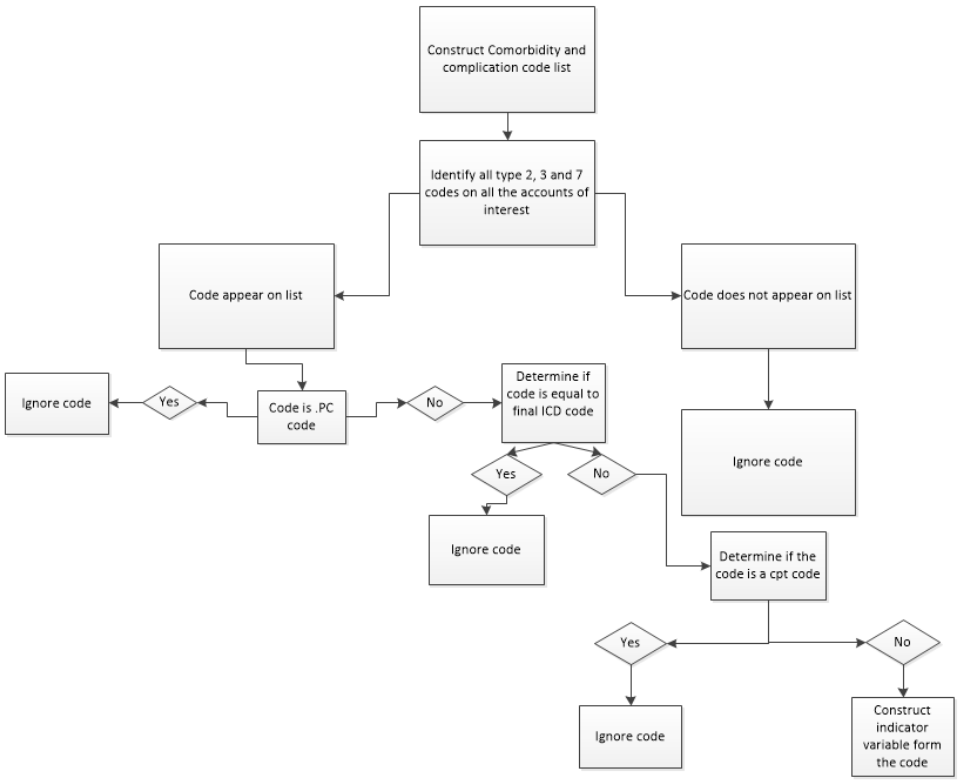

Diagram 2.1: Algorithm to determine if a code can act as a complication and comorbidity variable.

The training of the six decision trees transpires in SAS Enterprise miner. Table 2.4 indicate properties associated with the respective decision trees

Table 2.4: Properties of decision trees

Model	Decision tree
Training data set proportion	50 % of 50 000 observations
Explanatory variables	ICD codes
Response variables	Binary variable indicating extended LOS
Minimum leaf size	50
Minimum split size	50
Stopped training	Assessment based on misclassification rate

Additionally, Figure 2.1 provides an illustration of the structure of one of the decision trees involved in the process. Figure 2.2 provides an indication of the training and validation misclassification rate trade − off. The model training stops at iteration 11 since overfitting starts to become evident at this iteration because the validation misclassification rate starts to deviate severely from the training misclassification rate.

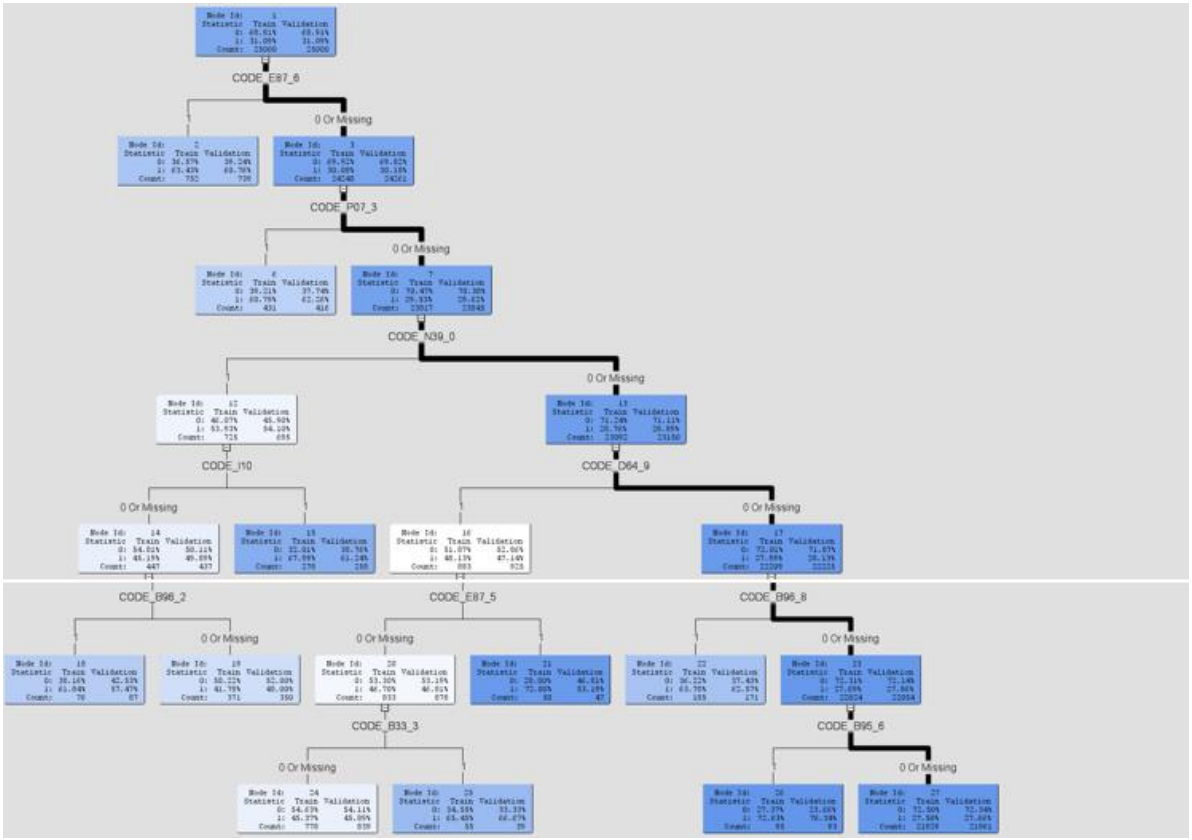

Figure 2.1: Decision tree indicating significant codes in predicting extended accommodation

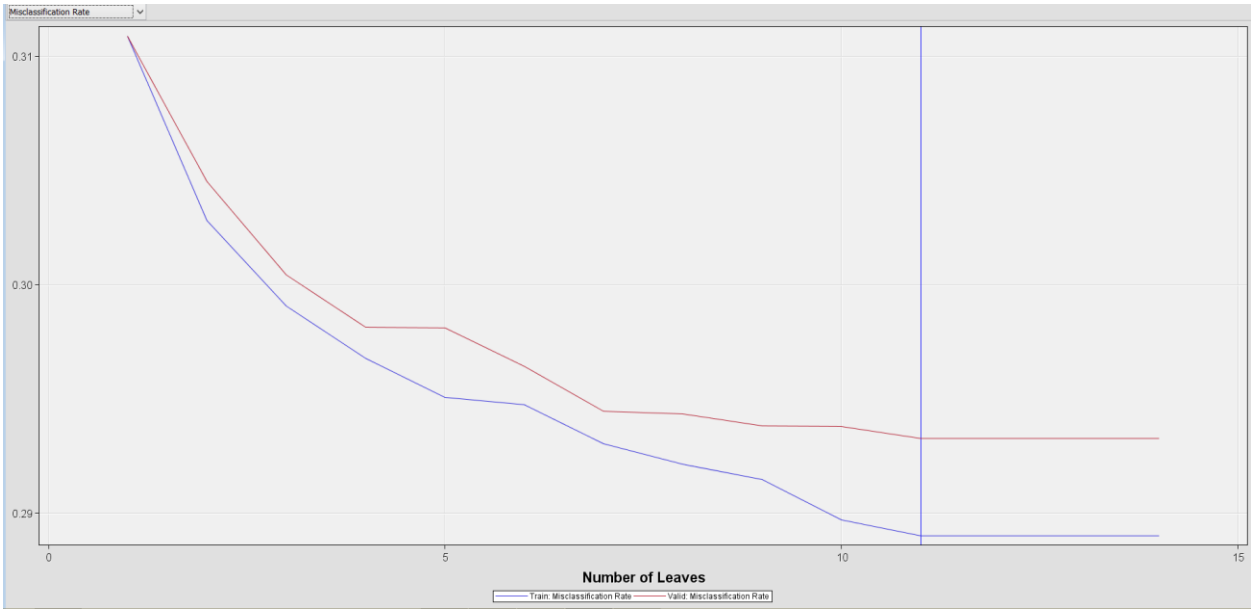

Figure 2.2: Training and validation trade-off for the decision tree in Figure 2.1

2.4 DISTRIBUTION OF DATA

One of the most common characteristics of healthcare financial data is the skewness present. Another prominent observation is that, on average, a readmission is more expensive than an index admission. This is visible in Figure 2.3. The mean of the readmissions (filled circle in the boxplot on the right) is higher in comparison with the index admissions (filled circle in the boxplot on the left). This provides further motivation to investigate methods to reduce the occurrence of readmissions, as well as, assist in constructing a cost matrix, as Chapter 4 will illustrate.

Also visible in Figure 2.3 is the right skewness of the data in the fact that the mean is higher than the median (black solid horisontal line) in both plots.

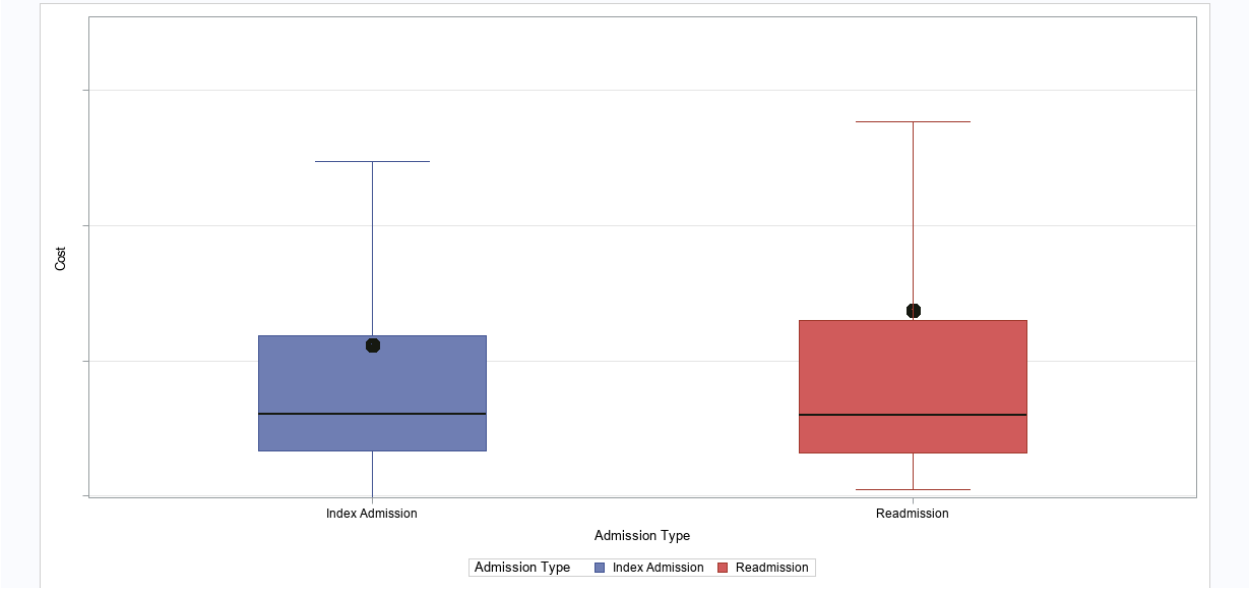

Figure 2.3: Distribution of Index admission versus Readmission (black solid line – median, filled circle – mean)

2.5 SUMMARY

This chapter provided significant insight in the data preparation process. It displayed the application of machine learning to solve problems entailing data management. In addition, this study believes that improved results can be obtained at the modelling stage if thorough data preparation and feature engineering are conducted. It is important to realise that in practice a data scientist is rarely presented with a dataset that is perfect to start modelling immediately.

Time spent on data querying, as well as, variable construction is a crucial part of the process and significant headway is possible if the data scientist thinks creatively in this regard. The chapter also illustrated ways to reduce the number of input variables. This reduction in input variables is continued in Chapter 4. Also highlighted in this chapter is the value of approaching industry experts to assist the data scientist in making informed decisions prior to modelling.

CHAPTER 3
DESCRIPTION OF ALGORITHMS, MODELS AND RELATED CONCEPTS

3.1 INTRODUCTION

Various algorithms exist in the literature that can model, either categorical responses, or continuous responses based on observed input variables. The algorithms vary in complexity and interpretability, as well as, whether it models the response in a linear or nonlinear fashion. The training time the different algorithms require to reach a convergence criterion vary substantially, but this is generally of less importance.

Overfitting tend to be a likely phenomenon most algorithms can fall victim to. Therefore, methods to avoid overfitting receive ample attention in machine learning literature. From experience this study believes that in practice there is often discord regarding the preference between complexity and interpretability. In some instances, complex techniques, also referred to as black – box techniques, can potentially provide a better solution but due to its complex nature stakeholders tend to distrust the generated results and prefer a more intuitive technique, for example decision trees. Occasionally less complicated techniques can outperform techniques that are more complex, as Chapter 4 and Chapter 5 illustrates.

This chapter focus on the advantages and disadvantages of four techniques, namely, logistic regression, decision trees, neural networks and SVM. Logistic regression and decision trees are relatively simple, whereas neural networks and SVM's are considered more complex. However, all the techniques have underlying theory and this chapter aims to summarise the important concepts with regards to each technique in a simple manner.

3.2 LOGISTIC REGRESSION

3.2.1 Introducing logistic regression

Logistic regression is a parametric modelling technique and is a member of the family of generalised linear models (GLM). The two main aspects associated with logistic regression that are in contrast to ordinary linear regression, are the introduction of a link function and the restriction to solely categorical response variables.

3.2.2 Theory of logistic regression

Logistic regression is a common machine learning technique. Hastie, Tibshirani and Friedman (2008: 119) state modelling the posterior probabilities as a linear function of the input features as the objective of logistic regression. Estimation of the coefficients β proceeds as described in this section.

If the response variable consists of K classes, $K-1$ logistic regression functions are required (Hastie *et al.*, 2008: 119). Considering the categorical dependant variable G, a probability for classification into class k, $k=1,\ldots,K$ is of interest with coefficients β_{k0} and $\underline{\beta}_k$ influential in determining the probability. In order to guarantee that the probabilities sum to one, the log odds are modelled linearly in the inputs $X=\underline{x}$ (Hastie *et al.,* 2008: 119),

$$\begin{aligned}
\log\frac{P(G=1|\,X=\underline{x})}{P(G=K\,|\,X=\underline{x})} &= \beta_{10}+\underline{\beta}_1^T\underline{x}\\
\log\frac{P(G=2\,|\,X=\underline{x})}{P(G=K\,|\,X=\underline{x})} &= \beta_{20}+\underline{\beta}_2^T\underline{x}\\
&\vdots\\
\log\frac{P(G=K-1|\,X=\underline{x})}{P(G=K\,|\,X=\underline{x})} &= \beta_{(K-1)0}+\underline{\beta}_{K-1}^T\underline{x}
\end{aligned} \tag{3.1}$$

In order to see that the probabilities in (3.1) sum to one, consider the case of $K=2$

From (3.1)

$$\log\frac{P(G=1|\,X=\underline{x})}{P(G=2\,|\,X=\underline{x})}=\beta_{10}+\underline{\beta}_1^T\underline{x} \tag{3.2}$$

$$\therefore P(G=1|\,X=\underline{x})=\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})\times P(G=2\,|\,X=\underline{x}) \tag{3.3}$$

But $$P(G=1|\,X=\underline{x})+P(G=2\,|\,X=\underline{x})=1$$

Thus $$P(G=2\,|\,X=\underline{x})=1-P(G=1|\,X=\underline{x}) \tag{3.4}$$

From (3.3) and (3.4) $P(G=1|\,X=\underline{x})=\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})\times(1-P(G=1|\,X=\underline{x}))$

$$\therefore P(G=1|\,X=\underline{x})=\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})-\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})P(G=1|\,X=\underline{x}))$$

$$\therefore P(G=1|\,X=\underline{x})+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})P(G=1|\,X=\underline{x})=\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})$$

$$\therefore P(G=1|\,X=\underline{x})(1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x}))=\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})$$

$$\therefore P(G=1|\ X=\underline{x})=\frac{\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})} \tag{3.5}$$

$$\therefore 1-P(G=1|\ X=\underline{x})=1-\frac{\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}$$

From (3.4)
$$P(G=2\,|\,X=\underline{x})=\frac{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}-\frac{\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}$$

$$\therefore P(G=2\,|\,X=\underline{x})=\frac{1}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})} \tag{3.6}$$

The sum of (3.5) and (3.6) is

$$P(G=1|\ X=\underline{x})+P(G=2\,|\,X=\underline{x})=\frac{\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}+\frac{1}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}$$

$$=\frac{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}{1+\exp(\beta_{10}+\underline{\beta}_1^T\underline{x})}$$

$$=1$$

Agresti (2002: 166) affirms that for a binary dependant variable Y the model of interest for a single independent variable x is

$$P(Y=1|\ X=x)=\pi(x)=\frac{\exp(\alpha+\beta x)}{1+\exp(\alpha+\beta x)} \tag{3.7}$$

from (3.7) it follows that

$$\frac{1-\pi(x)}{\pi(x)}=\frac{1-\dfrac{\exp(\alpha+\beta x)}{1+\exp(\alpha+\beta x)}}{\dfrac{\exp(\alpha+\beta x)}{1+\exp(\alpha+\beta x)}}=\frac{1+\exp(\alpha+\beta x)}{\exp(\alpha+\beta x)}-1$$

$$=\frac{1+\exp(\alpha+\beta x)}{\exp(\alpha+\beta x)}-\frac{\exp(\alpha+\beta x)}{\exp(\alpha+\beta x)}$$

$$=\frac{1}{\exp(\alpha+\beta x)}$$

$$\Rightarrow \frac{\pi(x)}{1-\pi(x)}=\exp(\alpha+\beta x) \tag{3.8}$$

Finally (Agresti 2002: 166),

$$\text{logit}[\pi(\text{x})] = \log \frac{\pi(\text{x})}{1-\pi(\text{x})} = \alpha + \beta x \tag{3.9}$$

Agresti (2002: 182) points out that the univariate case provided in (3.9) can be extended to multiple logistic regression for $\underline{x}$

$$\text{logit}[\pi(\underline{\text{x}})] = \log \frac{\pi(\underline{\text{x}})}{1-\pi(\underline{\text{x}})} = \alpha + \beta_1 x_1 + ... + \beta_p x_p \tag{3.10}$$

Hastie *et al.* (2008: 121) explains that a logistic regression model has the competency to quantify the influence of the input variables on the outcome. Thus, the interpretability associated with logistic regression marks its popularity. Assume x is continuous. Agresti (2002: 166) states that the sign of the β coefficient provides an indication of the increase (positive β) or decrease (negative β) in $\pi(\text{x})$ as x increases. What is more informative is the interpretation of the quantity e^{β}.

From (3.8)

$$\begin{aligned} \frac{\frac{\pi(x+1)}{1-\pi(x+1)}}{\frac{\pi(x)}{1-\pi(x)}} &= \frac{\exp(\alpha+\beta(x+1))}{\exp(\alpha+\beta x)} \\ &= \exp(\alpha+\beta(x+1)-\alpha-\beta x) \\ &= \exp(\beta) \end{aligned}$$

$$\Rightarrow \frac{\pi(x+1)}{1-\pi(x+1)} = \frac{\pi(x)}{1-\pi(x)} \exp(\beta) \tag{3.11}$$

where (3.11) illustrates the multiplicative effect, of magnitude e^{β}, on the odds for an one unit increase in x (Agresti 2002: 166).

Alternatively, from (3.9)

$$\log \frac{\pi(\text{x+1})}{1-\pi(\text{x+1})} - \log \frac{\pi(\text{x})}{1-\pi(\text{x})} = \alpha + \beta(x+1) - \alpha - \beta x = \beta$$

It is evident that an increase of one unit in x, results in an additive increase of β on the logit scale, thus the log of the odds increase additively with β for a one unit increase in x.

Generally, as stated in (3.10), it is useful to include several explanatory variables in the logistic regression model and not only a single explanatory variable. This leads to a multiple logistic regression model,

$$\text{logit}[\pi(\underline{x})] = \log\frac{\pi(\underline{x})}{1-\pi(\underline{x})} = \alpha + \beta\underline{x} = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p \tag{3.12}$$

Alternatively, as an extension of equation (3.7)

$$\pi(\underline{x}) = \frac{\exp(\alpha + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\alpha + \beta_1 x_1 + \cdots + \beta_p x_p)} \tag{3.13}$$

Agresti (2002: 183) explains the interpretation of the coefficient β_i as a quantification of the additive effect a change in variable x_i has on the log odds of observing an event $(Y=1)$ while keeping the other variables fixed. More precisely, $\exp(\beta_i)$ provides the quantity with which the odds of observing an event increases, multiplicatively, for each one − unit increase in the variable x_i while keeping the other variables fixed (Agresti 2002: 183).

The estimation of the parameters transpires through maximum likelihood (Hastie *et al.,* 2008: 120). Rice (2007: 267) describes the maximum likelihood estimate of a parameter, say θ, as the value of θ that will result in the maximisation of the likelihood, or stated differently, it is the value of θ that is as likely as possible to result in the generation of the observed data. According to Rice (2007: 267) the likelihood function for parameter θ given observations $x_1, x_2 \cdots, x_n$ is

$$lik(\theta) = f(x_1, x_2 \cdots, x_n \mid \theta) \tag{3.14}$$

Rice (2007: 268) states further, that for i.i.d. X_i

$$lik(\theta) = \prod_{i=1}^{n} f(X_i \mid \theta) \tag{3.15}$$

In addition, it is preferable to maximise the log likelihood (Rice 2007: 268)

$$l(\theta) = \sum_{i=1}^{n} \log[f(X_i \mid \theta)] \tag{3.16}$$

In the context of logistic regression, Hastie *et al.* (2008: 120) provides (3.17) as the log likelihood function

$$l(\underline{\beta}) = \sum_{i=1}^{N} \{y_i \log p(\underline{x}_i; \underline{\beta}) + (1-y_i)\log(1-p(\underline{x}_i; \underline{\beta}))\} \tag{3.17}$$

For a binary response, if all the inputs $\underline{x}_i, i = 1,...,n$ are unique, then $n = N$ (Agresti 2002: 192). In this case $p(\underline{x}_i; \underline{\beta})$ will be the Bernouli mass function. Alternatively, the N groups of identical inputs among $\underline{x}_i, i = 1,...,n$ form part of N independent binomial trails $Y_i \sim Bin(n_i, \pi_i)$ and consequently $p(\underline{x}_i; \underline{\beta})$ is the Binomial mass function.

The estimation process continues by differentiating equation (3.17) with respect to $\underline{\beta}$ followed by equating the formula, resulting from the differentiation, to zero (Hastie *et al.,* 2008: 120). To solve this equation, the Newton – Raphson algorithm is implemented (Hastie *et al.,* 2008: 120).

As explained by Hastie *et al.* (2008: 120), the following formula is implemented as part of the Newton – Raphson algorithm,

$$\underline{\beta}^{new} = \underline{\beta}^{old} - \left(\frac{\partial^2 l(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}^T} \right)^{-1} \left(\frac{\partial l(\underline{\beta})}{\partial \underline{\beta}} \right) \tag{3.18}$$

For the Newton – Raphson algorithm to commence, an initial value must be specified for $\underline{\beta}^{old}$ which usually is chosen as $\underline{0}$ (Hastie *et al.,* 2008: 121). Chapter 3.3 provides an illustration of the Newton – Raphson algorithm.

3.2.3 Advantages and disadvantages

Advantages**:**

- Inference on variables are possible (Hastie *et al.,* 2008: 121).
- Provides for interpretability of significant variables (Hastie *et al.,* 2008: 121).
- Have regularisation abilities, namely, Ridge regression and the Lasso (Hastie *et al.,* 2008: 61, 68).
- Low variance (stable).
- Wide use in practice creates confidence among stakeholders in the results generated by the model.

Disadvantages:

- High bias (logistic regression model attains a predefined structure) (Christie et al. 2015: 4-68).
- Although the Newton – Raphson algorithm generally converges, convergence is not certain (Hastie *et al.,* 2008: 121). In the case of nonconvergence, most software packages will have a stopping clause, for example, a maximum number of iterations.

- In the event of correlation between input variables, unexpected results regarding the coefficient sign of the variables, or the significance itself, may occur among the correlated variables (Hastie *et al.,* 2008: 122).

Chapter 3.7 describes regularisation and the bias – variance trade – off.

3.2.4 Logistic regression and SAS Enterprise Miner

SAS Enterprise Miner can seamlessly construct a logistic regression model. SAS Enterprise Miner allows changes to several model properties in order to strive towards an optimal model. One of these properties is the choice of variable selection techniques such as stepwise, forward and backwards selection (Christie *et al.,* 2015: 4-24 – 4-28). It is also possible to change the selection criterion for variables entering and exiting the model during the variable selection process (Christie *et al.,* 2015: 4-37). Several model selection statistics calculated on the validation data are available to base the selection of the final model on, for example misclassification rate (Christie *et al.,* 2015: 4-36). Christie *et al.* (2015: 4-10) points out the danger of loss in data due to missing values since by default only cases with no missing values across all variables are used in the estimation process. SAS Enterprise Miner can address the effect of skewness in the distribution of the input features by means of application of transformations on the input features (Christie *et al.,* 2015: 4-46). Christie *et al.* (2015: 4-69) mentions that as an attempt to mitigate the possible bias associated with logistic regression due to its predefined linear structure, SAS Enterprise Miner allows implementation of polynomial terms in the model which consist of combinations of input features, for example combining two input features by the product of the two input features $x_1 \times x_2$

3.3 EXAMPLE OF APPLICATION OF NEWTON – RAPHSON ALGORITHM

Agresti (2002: 144) provides the formulations (3.19), (3.20), (3.21) regarding the Newton – Raphson algorithm:

Let $$\underline{u} = (\partial l(\underline{\beta}) / \partial \beta_1, \partial l(\underline{\beta}) / \partial \beta_2, \ldots) \quad (3.19)$$

and *Hessian matrix* H with elements $h_{ab} = \partial^2 l(\underline{\beta}) / \partial \beta_a \partial \beta_b$ (3.20)

where $\underline{u}^{(t)}$ and $H^{(t)}$ are $\underline{u}$ and H in the point $\underline{\beta}^{(t)}$ where $\underline{\beta}^{(t)}$ is the possible value of $\hat{\underline{\beta}}$ at step t (Agresti 2002: 144).

At the t^{th} step, the iterative procedure approximates $l(\underline{\beta})$ near the point $\underline{\beta}^{(t)}$ by the terms that occur as part of $l(\underline{\beta})$'s second order Taylor series expansion (Agresti 2002: 144)

$$l(\underline{\beta}) \approx l(\underline{\beta}^{(t)}) + \underline{u}^{(t)'}(\underline{\beta} - \underline{\beta}^{(t)}) + \frac{1}{2}(\underline{\beta} - \underline{\beta}^{(t)})' H^{(t)}(\underline{\beta} - \underline{\beta}^{(t)}) \tag{3.21}$$

Subsequently, solve $\partial l(\underline{\beta}) / \partial(\beta) \approx \underline{u}^{(t)} + H^{(t)}(\underline{\beta} - \underline{\beta}^{(t)}) = \underline{0}$ for $\underline{\beta}$ to obtain the next possible value of $\hat{\underline{\beta}}$ which can be written as (Agresti 2002: 144)

$$\underline{\beta}^{(t+1)} = \underline{\beta}^{(t)} - (H^{(t)})^{-1}\underline{u}^{(t)} \tag{3.22}$$

Similar to Agresti (2002: 144), this section demonstrates the Newton – Raphson algorithm by considering an example of which the true maximum likelihood estimate is known. Agresti (2002: 144) considers the $bin(n, \pi)$ distribution,

$$P(Y = y) = \binom{n}{y}\pi^{y}(1-\pi)^{n-y}, \quad y = 0,1,2,\ldots,n \tag{3.23}$$

From (3.16)

$$\begin{aligned} l(\pi) &= \log(\pi^{y}(1-\pi)^{n-y}) \\ &= y\log(\pi) + (n-y)\log(1-\pi) \\ l'(\pi) &= \frac{y}{\pi} + \frac{y-n}{1-\pi} \\ &= \frac{(1-\pi)y + \pi(y-n)}{\pi(1-\pi)} \\ &= \frac{y - y\pi + y\pi - \pi n}{\pi(1-\pi)} \\ &= \frac{y - \pi n}{\pi(1-\pi)} \\ &= u \end{aligned}$$

From (3.20)

$$H = l''(\pi) = u' = -\frac{y}{\pi^{2}} + \frac{y-n}{(1-\pi)^{2}}$$

Choose $\pi^{(0)} = \frac{1}{2}$ then from (3.22), the first iteration is

$$\begin{aligned}\pi^{(1)} &= \pi^{(0)} - (H^{(0)})^{-1}u^{(0)} \\ &= \frac{1}{2} - \left[-\left(\frac{y}{\frac{1}{4}} \right) + \frac{y-n}{(1-\frac{1}{2})^2} \right]^{-1} \times \left(\frac{y-\frac{n}{2}}{\frac{1}{4}} \right) \\ &= \frac{1}{2} - \left[-4y + 4(y-n) \right]^{-1} \times 4(y-\frac{n}{2}) \\ &= \frac{1}{2} + \frac{1}{4n} \times (4y-2n) \\ &= \frac{1}{2} + \frac{y}{n} - \frac{1}{2} \\ &= \frac{y}{n}\end{aligned}$$

$\pi^{(1)} = \frac{y}{n}$ thus from (3.22), the second iteration is

$$\begin{aligned}\pi^{(2)} &= \pi^{(1)} - (H^{(1)})^{-1}u^{(1)} \\ &= \frac{y}{n} - \left[-\left(\frac{y}{\frac{y^2}{n^2}} \right) + \frac{y-n}{(1-\frac{y}{n})^2} \right]^{-1} \times \left(\frac{y-y}{\frac{y}{n}(1-\frac{y}{n})} \right) \\ &= \frac{y}{n}\end{aligned}$$

Therefore, since convergence has been reached the algorithm stops and $\hat{\pi} = \frac{y}{n}$

The chosen initial value plays a major part in the number of iterations that the algorithm requires to reach convergence. Suppose in the example above the initial value is chosen to be $\frac{y}{n}$ then the algorithm would have converged after a single step. Suppose an initial value of $\frac{1}{3}$ is chosen then the algorithm will converge at an iteration > 2 (Agresti 2002: 145).

3.4 DESCISION TREES

3.4.1 Introducing decision trees

Decision trees are a nonparametric algorithm that aim to find the most optimal separation of the sample space by considering a set of input variables, as well as, a relevant response variable. The algorithm is intuitive and is a multipurpose technique since, not only can it aid as a model

itself, but can also act as variable selection and variable manipulation technique to other algorithms.

3.4.2 Theory of decision trees

Although decision trees are straightforward to understand, it is effective in its working (Hastie *et al.*, 2008: 305). Decision tree modelling boils down to optimally separating the sample space into regions (Hastie *et al*., 2008: 305). Within each rectangular region, a model is fit (Hastie *et al.*, 2008: 305). The model is generally restricted to be only a constant (Hastie *et al.*, 2008: 305). Regression trees are used for continuous response variables whereas classification trees are used for categorical response variables (Hastie *et al*., 2008: 307 & 308). Hastie *et al*. (2008: 305 - 309) provides formulas (3.24) to (3.27) for decision trees. In this study, a classification tree is appropriate in order to model the binary outcome of manifestation of readmission after an index admission. Considering to which leaf the classification tree allocates a new observation to, the most common class residing in that leaf provides the predicted class of the new observation (Hastie *et al*., 2008: 309).

The proportion of observations in region (leaf) m that belongs to class k of the response variable (Hastie *et al*., 2008: 309).

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \varepsilon R_m} I(y_i = k) \tag{3.24}$$

where

$$x_i = input\ observation\ vector\ i$$

$$y_i = Response\ i$$

$$N_m = number\ of\ observations\ in\ region\ m$$

$$R_m = region\ m$$

$$k = class\ k\ of\ the\ response$$

and the class to which the tree classify an observation in leaf *m* is

$$k(m) = \arg\max_k \hat{p}_{mk} \tag{3.25}$$

whereas, for a continuous response

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{3.26}$$

where

$$c_m = the\ predicted\ value\ for\ leave\ m$$

$$x = Observation\ vector$$
$$R_m = region/leave\ m$$

with

$$\hat{c}_m = ave(y_i \mid x_i \in R_m) \tag{3.27}$$

The measures that play a role in the growth of a classification tree is misclassification, Gini index and cross – entropy (Hastie *et al*., 2008: 309). Regression trees usually utilise the squared error as part of a training algorithm to grow the tree (Hastie *et al*., 2008: 307). Several algorithms that grow decision trees exist, namely, Quinlan's ID3, C4.5 and CART (Marsland, 2009: 135).

3.4.3 Popularity of decision trees

Decision trees are a popular machine learning technique and can provide a way to perform variable selection prior to implementing another machine learning technique (De Ville & Neville, 2013: 4). Additionally, decision trees can aid as a mechanism to create variables which in turn can act as inputs to other machine learning techniques (De Ville & Neville, 2013: 4). Furthermore, decision trees can combine the levels of a categorical variable to form new grouping of levels (De Ville & Neville, 2013: 4). In Chapter 4, a discussion follows of how this study makes use of the preceding attributes of decision trees. The simple nature of the technique and consequently the ease in understanding by stakeholders also contributes to its popularity.

3.4.4 Usage, advantages and disadvantages

Further advantages of decision trees are the following:

- Circumvent the curse of dimensionality (see discussion in Chapter 3.7) due to its ability to disregard variables that the algorithm finds irrelevant (Christie *et al*., 2015: 3-28).
- Easy to understand by both analyst and stakeholders (De Ville & Neville 2015: 6).
- Ability to model nonlinear patterns in data and no distributional assumptions are required (Gordon, 2013: 1).
- Multiple types of target and input variables are plausible such as multi categorical, interval, binary and ordinal (Gordon, 2013: 1).
- Missing values pose no problem (Christie *et al.,* 2015: 3-30).
- Identification of interactions between variables (Gordon, 2013: 5).
- Easy to interpret (Hastie *et al.,* 2008: 305).
- Visual representation of the tree advances understanding of algorithm (Gordon, 2013: 5).
- Automatically divide the population into pockets which can be used in practice (Gordon, 2013: 5).

The disadvantages of decision trees are

- The automatic variable selection results in the inability to force certain variables into the model even though it is deemed not significant by the decision tree algorithm (Gordon, 2013: 6). SAS Enterprise Miner does possess the interactive tree growing functionality which overcomes this particular disadvantage (Gordon, 2013: 6).
- The simple nature of decision trees are in some instances abused by feeding the model unrelated variables without a logical thought process behind the inclusion of the particular variable (Lemon *et al*. cited in Gordon, 2013: 6). Without the necessary caution this can result in identifying unrelated variables as significant by chance.
- High variance, unstable model. Therefore, if a decision tree is trained on two different samples from the same population the two resulting decision trees might differ in structure and in prediction (Gordon, 2013: 6). Hastie *et al*. (2008: 312) also mentions the fact that decision trees are unstable (high variance). A pair of decision trees trained separately on two different samples will likely produce noticeably different results. Hastie *et al*. (2008: 312) proposes *bagging* (combining multiple decision trees into one model) as a possible remedy.
- The contour plot (perspective drawing) of the predicted function will typically be unsmooth (consist of abrupt up and down changes in the surface) which can lead to an increase in bias *(Hastie et al.,* 2008: 312). Hastie *et al.* (2008: 312) especially warn against possible bias resulting from an unsmooth prediction surface proposed by the decision tree in the regression context. This is less of a concern in the setting of modelling a categorical response (Hastie *et al.* 2008: 312).
- Categorical variables with multiple categories can incentivise overfitting by finding a split in one of the many possible ways to split the variable significant on the data by chance (Hastie *et al.,* 2008: 310).

3.4.5 Decision trees and SAS Enterprise Miner functionality

SAS Enterprise Miner inherently grows decision trees by performing multiple hypothesis tests (Christie *et al*., 2015: 3-29). Considering a binary response variable, the algorithm performs a hypothesis test H_0 : Independence (Christie *et al*., 2015: 3-29) at each candidate splitting point. In other words, the null hypothesis is that there is no association between the particular side of the split point an observation finds itself and the value of the observations' response.

For continuous input variables, every unique value can potentially end up being the chosen split point (Christie *et al*., 2015: 3-29). For example, suppose the input variable is *age* consisting of integers ranging between 0 and 100, the significance of the variable age at each integer within [0; 100] is tested. For categorical input variables, the procedure tests for the significance of the

variable for all possible divisions of the categories into, by default, two groups. The p – values resulting from the hypothesis tests are transformed to what is referred to as the *logworth* (Christie *et al*., 2015: 3-29),

$$\log worth = -\log(p - value) \tag{3.28}$$

Subsequently, per input variable, the algorithm considers all *logworths* resulting from the hypothesis tests and the splitting point having the largest *logworth* is considered the best split for that particular input variable (Christie *et al*., 2015: 3-29). However, for each input variable at least one candidate split point must exceed a predefined threshold, in order for the variable to be considered as a candidate spitting variable (Christie *et al*., 2015: 3-29). Thus, for all qualified splitting variables, the algorithm determines the optimal split for each input variable and subsequently the variable whose optimal split produces the highest *logworth* will act as the first split of the decision tree (Christie *et al*., 2015: 3-32). This process repeats itself in each of the resulting nodes of the first split (Christie *et al*., 2015: 3-33).

This process will lead to what is called the *maximal tree* (Christie *et al*., 2015: 3-36). The *maximal tree* is the resulting tree after all variables, which satisfy the *logworth* threshold, has been split upon at its optimal split point (Christie *et al*., 2015: 3-36). The *maximal tree* is expected to have poor generalisation ability on new data due to overfitting on the training data (Christie *et al*., 2015: 3-36). In order to avoid the phenomenon of overfitting the algorithm sequentially eliminates leaves from the *maximal tree* (Christie *et al*., 2015: 3-56). The algorithm investigates the performance of all trees resulting from the removal of one variable split form the *maximal tree* and chooses the subtree with the best performance on the validation data (Christie *et al*., 2015: 3-56). The process continues, by investigating the performance on the validation data by removing another split from the subtree deemed the best during the first iteration of pruning, as described above. Consequently, the algorithm identifies the subtree with the best performance on the validation data after two splits have been pruned (Christie *et al*., 2015: 3-59). The process continues in a similar way, removing more splits at each iteration (Christie *et al*., 2015: 3-59). Considering the identified best subtrees for each iteration, the final model will be the model that among all subtrees is the simplest yet provides the best performance on the validation data (Christie *et al*., 2015: 3-60).

Additional constraints and specifications:

- The minimum number of observations each node must hold in order to be considered to split further (Christie *et al*., 2015: 3-30).
- The minimum number of observations each resulting node from a split must contain (Christie *et al*., 2015: 3-30). If the resulting nodes do not meet this requirement, the algorithm prunes the split that generated the nodes.

- Application of the Bonferroni correction to avoid attaining significant splits by chance (Christie *et al.*, 2015: 3-30). For this reason Hastie *et al*. (2008: 310) warns against the use of variables with several category levels and also states that it poses a risk for overfitting if such a variable split by chance.

3.5 SUPPORT VECTOR MACHINES

3.5.1 Introducing support vector machines

The SVM is a machine learning technique that searches for a separating hyperplane which is not restricted to be linear nor to perfectly separate the classes of the response variable. That is, it allows, to a certain extent, training cases to reside on the wrong side of the decision boundary.

3.5.2 Perceptron and optimal separating hyperplane

The perceptron and optimal separating hyperplane are two methods with the ability to establish boundaries to separate linearly separable data into regions (Hastie *et al*., 2008: 129). Hastie *et al*. (2008: 129) mention that the perceptron and optimal separating hyperplane provide the fundamental background to the concept of support vector classifiers. The perceptron algorithm searches for a hyperplane (linear decision boundary) by minimising the distance between misclassified points and the proposed hyperplane (Hastie *et al*., 2008: 130). Whereas the optimal separating hyperplane (linear decision boundary) attempts to maximise the distance between the point(s) closest to the hyperplane (on either side) and the hyperplane itself (Vapnik 1996 cited in Hastie *et al*., 2008: 132). The disadvantage of both the former and the latter is the fact that data must be, linearly separable, that is the data must be of such a nature that a plane (linear decision boundary) can separate the different output classes, perfectly (Hastie *et al*., 2008: 131 & 134).

Equation (3.29) formulates the concept of linearly separable data algebraically (Webb, 2002: 124),

$$\underline{v}' y > 0 \qquad \forall \ responses \ y_i \ corresponding \ to \ inputs \ x_i \tag{3.29}$$

where $\underline{v}$ is a vector containing the coefficients of the hyperplane.

Therefore, a formulation describing the perceptron algorithm is (Hastie *et al.*, 2009: 131):

Consider $(\underline{x}_i, y_i)$ where $\underline{x}_i$ is a p – dimensional input vector and $y_i \in (-1,1)$ indicating the class membership of each input vector. The goal of interest is to minimise

$$D(\underline{\beta}, \beta_0) = -\sum_{i \in M} y_i (\underline{x}_i^T \underline{\beta} + \beta_0) \tag{3.30}$$

where M is the set of misclassified inputs.

$D(\underline{\beta}, \beta_0)$ is always positive and is related to the distance between incorrectly classified inputs and the decision boundary (Hastie *et al.*, 2009: 131).

In order to grasp the fact that (3.30) is always positive, consider the following scenarios:

Let the true value of y_1 be $+1$.

$x_1 \in M$, meaning x_1 is misclassified by predicting the associated $\hat{y}_1$ to be -1, hence $(\underline{x}_1^T \underline{\beta} + \beta_0) < 0$, therefore equation (3.30) becomes,

$$-y_1(\underline{x}_1^T \underline{\beta} + \beta_0) = -(+1)(-1) = +1 > 0 .$$

Alternatively, let the true value of y_1 be -1

$x_1 \in M$, meaning x_1 is misclassified by predicting the associated $\hat{y}_1$ to be $+1$, hence $(\underline{x}_1^T \underline{\beta} + \beta_0) > 0$, therefore equation (3.30) becomes,

$$-y_1(\underline{x}_1^T \underline{\beta} + \beta_0) = -(-1)(+1) = +1 > 0 .$$

The optimal separating hyperplane bases its underlying machinery on a similar concept to the perceptron. As Hastie *et al*. (2009: 132) describes, the algorithm aims to maximise the margin $\mathbf{M}$ (distinguish $\mathbf{M}$ from the set of misclassified points M)

$$\max_{\underline{\beta}, \beta_0, \|\underline{\beta}\|=1} \mathbf{M} \tag{3.31}$$

$$\text{subject to } y_i(\underline{x}_i^T \underline{\beta} + \beta_0) \geq \mathbf{M}, i = 1, ..., N$$

As mentioned, the perceptron and optimal separating hyperplane require training data that is linearly separable, and this acts as a major drawback. An alternative to overcome this problem follows in the succeeding section.

3.5.3 Support vector classifier and support vector machine

The support vector classifier has the ability to produce a linear decision boundary on data that is not linearly separable (Hastie *et al*., 2009: 417). Therefore, the algorithm can tolerate, to a certain extent, observations occurring on the incorrect side of the decision boundary in the training data. This is accomplished by the introduction of slack variables (Hastie *et al*., 2009: 418 & 419).

Formula (3.31) suggests that the decision boundary with the largest margin $\mathbf{M}$ is of interest (Hastie *et al*., 2009: 132). This implies that the distance between the decision boundary and the closest observation to the decision boundary, is $\mathbf{M}$ (Hastie *et al*., 2009: 132). Hastie *et al*. (2009: 132 & 418 − 419) presents the following formulas illustrating the derivation of the optimal

separating hyperplane, as well as, how the optimal separating hyperplane can be generalised to form the support vector classifier.

Eliminate the restriction $\| \beta \| = 1$

$$\frac{1}{\| \underline{\beta} \|} y_i (\underline{x}_i^T \underline{\beta} + \beta_0) \geq \mathrm{M} \tag{3.32}$$

$$\Leftrightarrow \quad y_i (\underline{x}_i^T \underline{\beta} + \beta_0) \geq \mathrm{M} \| \underline{\beta} \| \tag{3.33}$$

set $$\| \underline{\beta} \| = \frac{1}{\mathrm{M}} \tag{3.34}$$

thus from (3.31)

$$\min_{\underline{\beta}, \beta_0} \frac{1}{2} \| \underline{\beta} \|^2 \tag{3.35}$$

subject to, from (3.33) and (3.34), $y_i (\underline{x}_i^T \underline{\beta} + \beta_0) \geq 1, i = 1, \ldots, N$

Now introduce the slack variable, ξ_i . The slack variables indicate points residing on the incorrect side of the decision boundary (Hastie *et al.*, 2009: 419).

According to Hastie *et al*. (2009: 419), the support vector classifier aims to minimise the same objective function as for the optimal separating hyperplane, as expressed in formula (3.35). However, the constraints associated with the support vector classifier differ from those of the optimal separating hyperplane (Hastie *et al.*, 2009: 419).

$$\begin{Bmatrix} y_i (\underline{x}_i^T \underline{\beta} + \beta_0) \geq 1 - \xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq \text{constant} \end{Bmatrix} \tag{3.36}$$

Finally, for the sake of mathematical convenience, Hastie *et al*. (2009: 420) proposes the following optimisations criterion,

$$\begin{Bmatrix} \min_{\underline{\beta}, \beta_0} \frac{1}{2} \| \underline{\beta} \|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{subject to } \xi_i \geq 0, \;\; y_i (\underline{x}_i^T \underline{\beta} + \beta_0) \geq 1 - \xi_i, \forall i \end{Bmatrix} \tag{3.37}$$

The cost parameter, C, influences the amount of regularisation applied in the training process (Hastie *et al.*, 2009: 421):

Small value for C $\Rightarrow$ allow several positive ξ_i $\Rightarrow$ a great deal of regularisation $\Rightarrow$ broad margin.

Large value for C $\Longrightarrow$ allow little positive ξ_i $\Longrightarrow$ little regularisation $\Longrightarrow$ narrow margin.

The fact that the support vector classifier allows points residing in the wrong side of the decision boundary, results in the support vector classifier acting superior to the optimal separating hyperplane. However, the support vector classifier remains restricted to be linear of nature. The SVM aims to outperform the support vector classifier by circumventing the restriction of linearity.

Hastie *et al*. (2009: 423) explains the extension of the support vector classifier, which provide a linear decision boundary with respect to the original input space, to the SVM, which provide a nonlinear decision boundary with respect to the original input space. The implementation of basis functions (e.g. polynomials or splines) on the original input space and subsequently fitting a support vector classifier in this transformed input space, results in a nonlinear decision boundary in the original input space, and thus also results in the construction of a SVM (Hastie *et al*., 2009: 423).

3.5.4 Usage, advantages and disadvantages

The SVM provide the opportunity to model nonlinear relationships in the original input space (Hastie *et al*. 2009: 423). The training process of the SVM is resource intensive due to the need to obtain the inverse of a data matrix, which can involve intense computation time, especially for enormous datasets (Marsland 2009: 119). The SVM is less intuitive in comparison with decision trees and logistic regression. However, the SVM has the potential to outperform other machine learning techniques in some instances, especially on datasets of moderate size (Marsland 2009: 119).

The functionality of SAS Enterprise Miner with regards to the SVM is restricted to a binary response variable and does not support categorical response variables consisting of more than two categories (*SAS Institute Inc*. 2018d). More flexibility is available with the input variables since any of the following types of input variables are supported by the SVM in SAS Enterprise Miner: binary (0/1), ordinal (categorical with a specific ordering), nominal (categorical with no specific ordering) and interval (continuous) (*SAS Institute Inc*. 2018d). As mentioned in Section 3.5.3 the SVM attains its nonlinear modelling capabilities by the introduction of kernel functions. SAS Enterprise Miner make use of kernel functions such as a linear function, polynomial kernel function, radial basis function or sigmoid function (*SAS Institute Inc*. 2018d). As is common in SAS Enterprise Miner, observations with missing values are not used during the training process except if specified by the user that missing values must be a level on their own (*SAS Institute Inc*. 2018d).

3.6 NEURAL NETWORKS

3.6.1 Introducing neural networks

A neural network is a non – linear machine learning technique and provides extensive versatility in terms of complexity by providing the opportunity to alter its architecture. The term architecture refers to the model structure of a neural network (SAS Institute Inc., 2018f). The training of the algorithm, as well as the predictions provided, are not as intuitive as other techniques but promises increased accuracy. The algorithm warrants informative research on how to improve the performance further or how to conduct inference on the results for better interpretation ability.

3.6.2 Theory of neural networks

Generally, the term neural network refers to the multi − layer perceptron (Christie *et al.,* 2015: 5-3). A neural network is usually proclaimed a cryptic predictive model (Christie *et al.,* 2015: 5-3). However, in essence, the relationship between a neural network and a regression model is clear (Christie *et al.,* 2015: 5-3). Hastie *et al*. (2008: 389) describes a neural network as the construction of linear combinations of the input variables which in turn undergo a non − linear transformation, leading to a prediction of the response variable. Neural networks can operate in both regression and classifications scenarios (Hastie *et al*., 2008: 392).

Consider Figure 3.1, a visual representation of an ordinary regression model in modelling a binary outcome,

$$y = b + W_1 X_1 + W_2 X_2 + W_3 X_3 \tag{3.38}$$

A neural network has an associated bias term, rather than using the term, *intercept*, that is commonly associated with an ordinary regression model (Christie *et al.,* 2015: 5-5). In addition, a neural network refers to the *parameter estimates*, which an ordinary regression model typically refer to as the estimated coefficients, as the *weights* of the neural network (Christie *et al.* 2015: 5-5). There exist similarities between Figure 3.1 and the visual representation of the neural network in Figure 3.2, except that in Figure 3.1 there is no *hidden units* and consequently no *hidden layers*. Therefore, Christie *et al.* (2015: 5-4) describes a neural network as a type of regression model fitted on a linear transformation of the original inputs, which is known as the *hidden units*. Brink (2018) also linked neural networks and regression by means of illustrations similar to the subsequent figures in Section 3.6.2.

The *hidden layer* of a neural network comprises of *hidden units.* The neural network in Figure 3.2 has three hidden units and one hidden layer. Whereas, the ordinary regression model illustrated in Figure 3.1 is fitted on the original inputs (X_1, X_2, X_3) and not on linear transformations of the original inputs (Z_1, Z_2, Z_3).

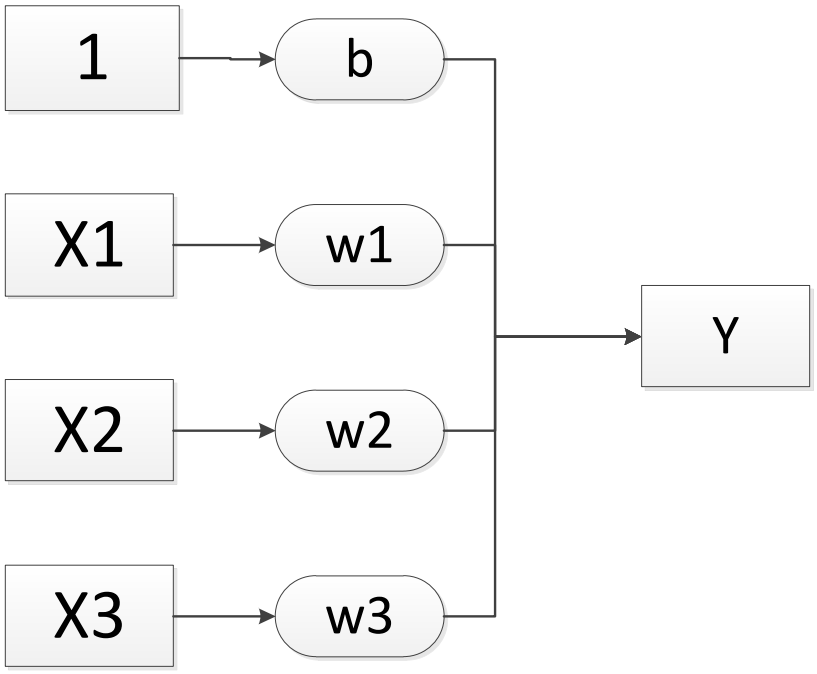

Figure 3.1: Graphical representation of a regression model, with coefficients b, w1, w2 and w3

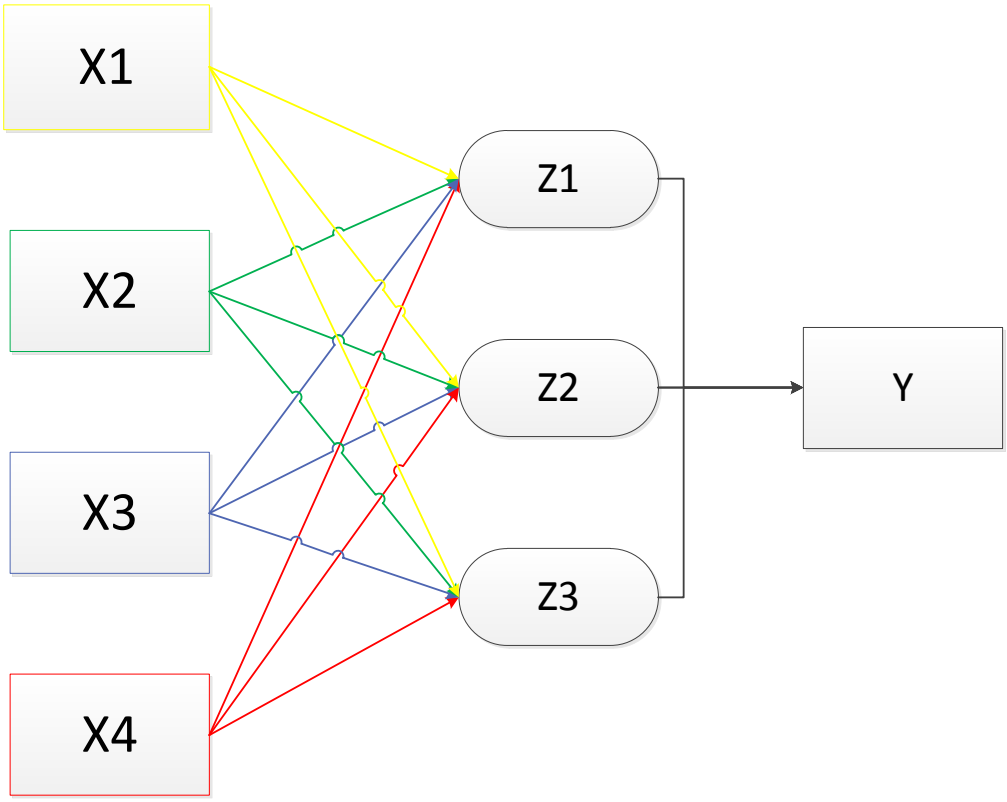

Figure 3.2: Graphical illustration of a single layer neural network

Figure 3.4 illustrates a neural network, with four *hidden units* and three outcome categories. This is the typical structure of a neural network applied in a classification context where the outcome consist of multiple classes (Hastie *et al*., 2008: 392). Each class represented by a binary indicator variable (Hastie *et al*., 2008: 392). Thus each outcome node represents the probability of classification into its associated class (Hastie *et al*., 2008: 392). Instead, Figure 3.2 represents the typical structure of a neural network in a regression context and thus have only one outcome node exposed to the identity transformation (Hastie *et al*., 2008: 392).

There is one hidden layer in Figure 3.4; however, multiple hidden layers are also possible (Hastie *et al*., 2008: 393). The discussion of the structure of the neural network in Figure 3.4, having one hidden layer and four hidden units, is based on the discussion of Hastie *et al*. (2008: 392 - 395) and lecture notes of Prof SJ Steel (Steel, 2017), based on Hastie *et al*. (2008).

Figure 3.4 illustrates an example of a neural network. The neural network consists of (for coefficients/weights α and β):

1. Five input variables, X_p, $p = 1,2,3,4,5$
2. Four hidden units, $Z_m, m = 1,2,3,4$
3. Three output variables, $Y_k, k = 1,2,3$
4. Transform the input variables linearly, $V_m = (\alpha_{om} + \underline{\alpha}_m{}^T \underline{X}), m = 1,2,3,4$

   Note: The full vector of inputs $\underline{X}$ are involved in calculating each $\underline{V}_m, m = 1,2,3,4$. This illustrates that each input contributes to each hidden unit. Thus no variable selection is involved.
5. The hidden units Z_m are nonlinear transformations of V_m

   $Z_m = \sigma(V_m), m = 1,2,3,4$

   σ is the activation function which is non − linear of nature
6. Transform Z_m linearly to obtain T_k

   $T_k = \beta_{0k} + \underline{\beta}_{0k}{}^T \underline{Z}, k = 1,2,3$
7. Transform the T_k's to output $Y_k = g_k(\underline{T}), k = 1,2,3$. The function g_k can be linear or non – linear.

In the event of a regression problem with one continuous output node, step (7) will typically consist of $Y = g(\underline{T})$ with the function g being the identity function (Hastie *et al*., 2008: 393).

In the application of a neural network in a single response binary classification problem, the logit function can act as the activation function σ (Christie *et al.,* 2015: 5-5). A popular function to use as the activation function σ is the sigmoid function (Hastie *et al*., 2008: 394)

$$\sigma(v) = \frac{1}{1 + \exp(-v)} \tag{3.39}$$

Another option for an activation function σ are the hyperbolic tangent function (Christie *et al.,* 2015: 5-4) and the soft – max function as function g (Marsland 2009: 58).

The activation function essentially determines if the *hidden unit* fires or not (Marsland 2009: 52). This means the activation function determines if the *hidden unit* will contribute to the rest of the network or not. In the case of the former, the magnitude of the contribution can be quantified by using a continuous activation function (see Figure 3.3) rather than a discontinuous activation function.

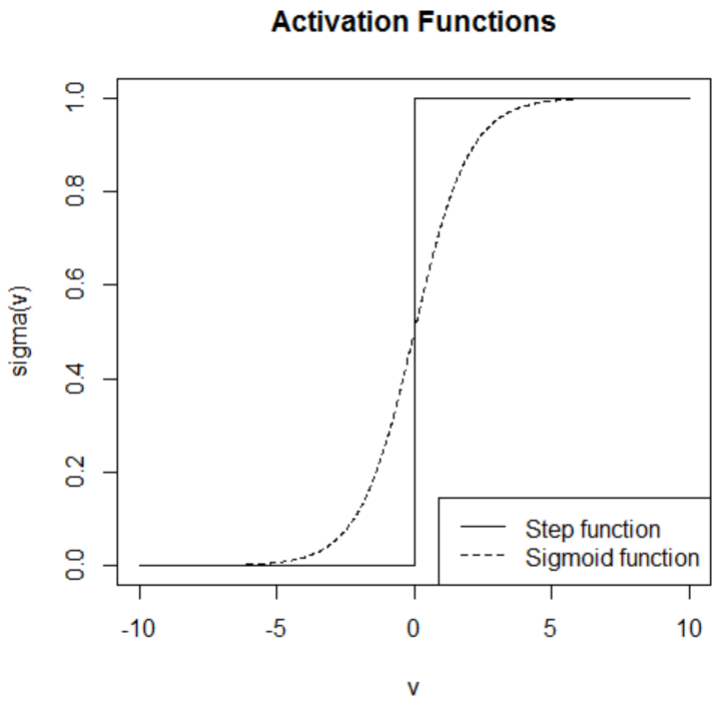

Figure 3.3: Activation functions

In order to see the relationship between ordinary regression and a neural network, consider how the components mentioned before can be adapted to represent a regression model, for k=3,

1. Five input variables, $X_p, p = 1,2,3,4,5$
2. No hidden units, $Z_m, m = 1,2,3,4$
3. Three output variables, $Y_k, k = 1,2,3$
4. Omit the linear transformation, $V_m = (\alpha_{om} + \underline{\alpha}_m{}^T \underline{X}), m = 1,2,3,4$
5. Omit the hidden units, Z_m, as nonlinear transformations of V_m

   $Z_m = \sigma(V_m),\ m = 1,2,3,4$
6. Transform X_p linearly to obtain T_k

   $T_k = \beta_{0k} + \underline{\beta}_{0k}{}^T \underline{X}, k = 1,2,3$
7. Transform T_k to output $Y_k = g_k(\underline{T}), k = 1,2,3$, where g_k is the identity function.

Finally, classify the observation to class k yielding the highest Y_k

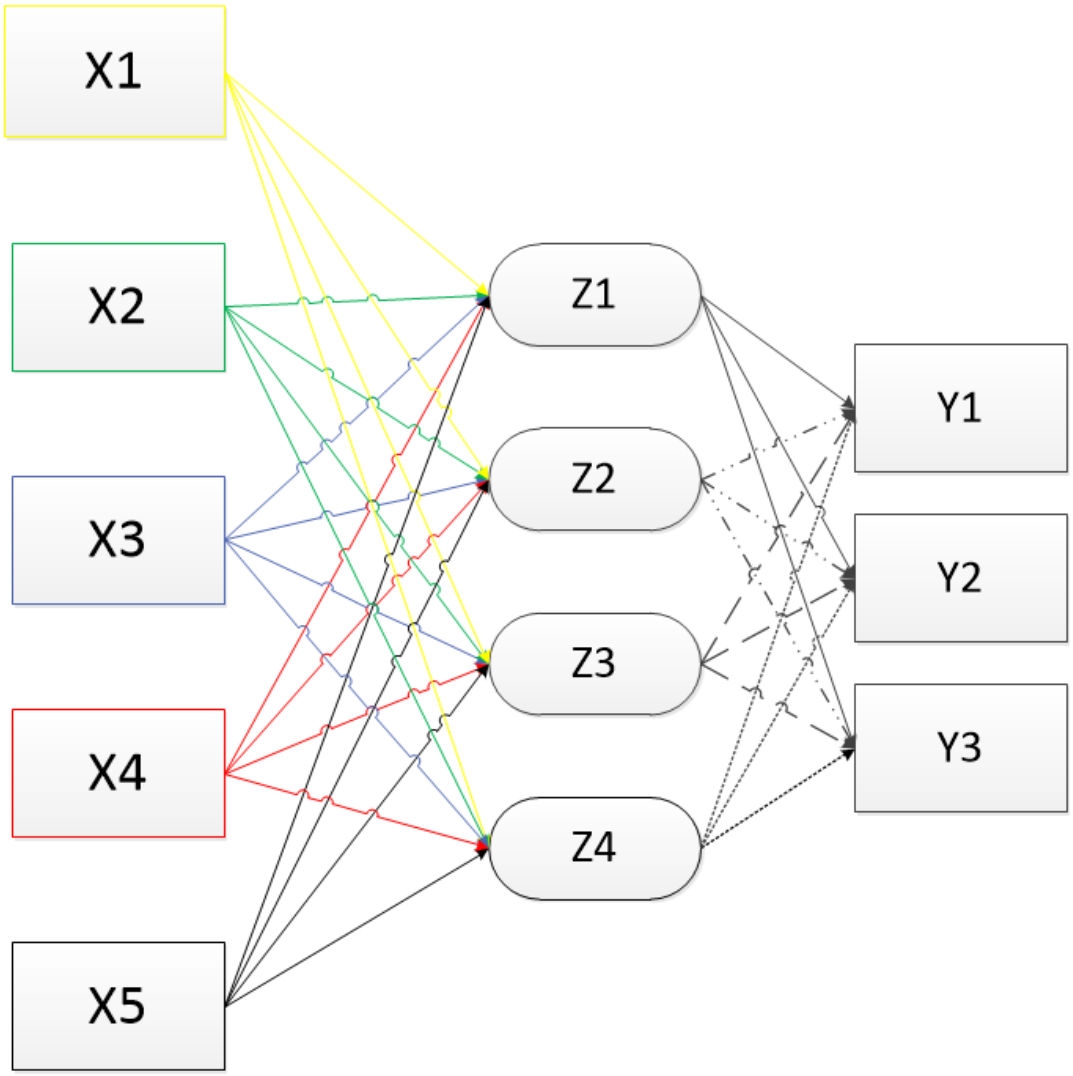

Figure 3.4: One − layer neural network

3.6.3 Optimisation and back propagation

Optimisation of model parameters generally involves the minimisation of an error function (Marsland, 2009: 247). The calibration of the weights by means of the back propagation algorithm relies on a process of first moving from the front to the back, thus forward, through the network, followed by moving from the back to the front, thus backwards, through the network (Marsland, 2009: 49). The process of moving forward through the network consists of the calculation of the outputs based on the observed inputs, intial chosen starting values for the weights (iteration 1) or the existing weights (iteration >1) (Marsland, 2009: 49). Subsequently, moving backwards entails that the weights are altered based on the error present in the output calculation, obtained by comparing the result of the latest forward process to the actual targets present in the training data (Marsland, 2009: 49).

Hastie *et al*. (2009: 395) mentions that the following weights needs to be estimated in the training process

$$\begin{aligned}&\{\alpha_{om}, \underline{\alpha}_m; m = 1, 2, ..., M\}, M(p+1) \, parameters \\ &\{\beta_{ok}, \underline{\beta}_k; k = 1, 2, ..., K\}, K(M+1) \, parameters\end{aligned} \tag{3.40}$$

where

M is the number of hidden nodes

K is the number of outputs

p is the number of inputs

Hastie *et al*. (2009: 395) goes further by specifying the error functions for both regression and classification.

Regression:

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(\underline{x}_i))^2 \tag{3.41}$$

Classification:

$$R(\theta) = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log f_k(\underline{x}_i) \tag{3.42}$$

During the training process the error function is minimised by the method of gradient descent (Mitchell, 1997: 97).

Figure 3.5 illustrates the process of back propagation in a simplified manner. The process flow is constructed based on the discussions of Hastie *et al*. (2009: 395 - 397), Mitchell (1997: 97) and Marsland (2009: 50 - 55).

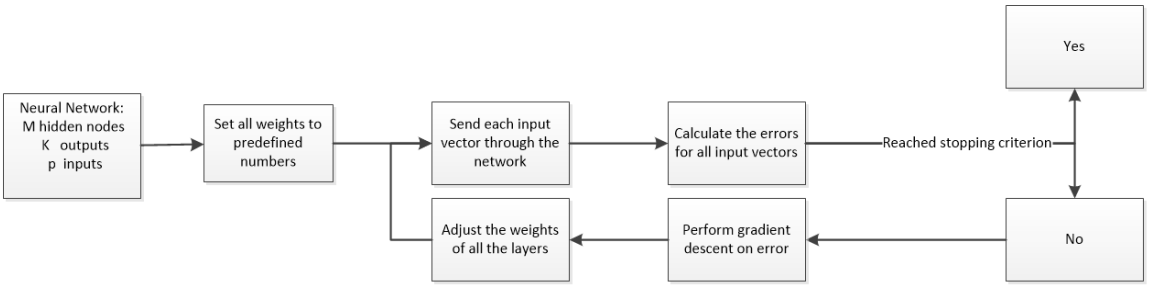

Figure 3.5: Back propagation illustrated visually

SAS Enterprise Miner utilises a process called s*topped training* (Christie *et al*., 2015: 5-21). This is the process where the resulting model at each iteration in the calibration of the neural network, acts as a possible model and, therefore, comparisons between the model at each iteration with models at subsequent and preceding iterations assist in determining which model is optimal (Christie *et al*., 2015: 5-21). This arises by considering a fit statistic associated with the model at each iteration on the validation dataset (Christie *et al*., 2015: 5-29). Training continues, even though the minimum of the fit statistic on the validation data is evident, until there is a trivial difference in the fit statistic on the training data for consecutive iterations or if the algorithm reaches the predefined maximum number of iterations (Christie *et al*., 2015: 5-29). The reason for this extension of the training is to ensure that the training of the neural network does not stop prematurely (Christie *et al*., 2015: 5-31). Furthermore, the looming danger of the algorithm being restrained to a local minimum is addressed by means of stopped training. Figure 3.6 illustrates the phenomenon of local and global minima.

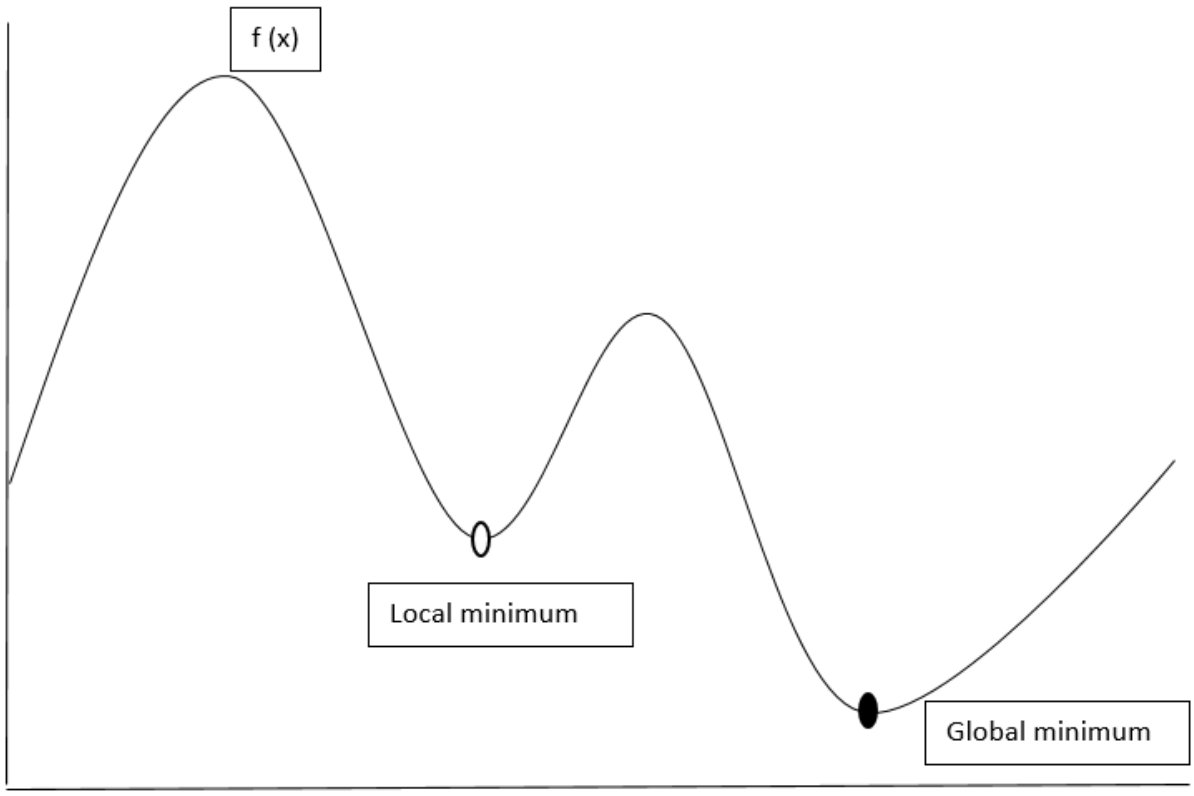

Figure 3.6: Local and Global minima

3.6.4 Usage, advantages and disadvantages

One of the main benefits of neural networks is the ability to model outcomes which is nonlinear functions of the inputs (Mitchell, 1997: 95). The capability of a neural network to model non – linear patterns can be destroyed if the network architecture does not include non – linear activation functions. (Mitchell, 1997: 96) points out that a network consisting of multiple layers

where each layer consists only of a linear transformation of the output from the previous layer, results in a linear model, even though multiple layers are involved.

For example, consider vector $\underline{x} = (1,4,7)$. Suppose in the first hidden layer $\underline{x}$ is transformed to $\underline{z}$ as follows, $\underline{z} = 2\underline{x} + 3 = (5,11,17)$. Subsequently, transform $\underline{z}$ linearly to $\underline{y} = 5\underline{z} - 1 = (24,54,84)$. It is possible to perform both these transformation in a single step, $\underline{y} = 10\underline{x} + 14$, which is still a linear function. Therefore, nothing is gained by the addition of multiple linear steps (graphically represented by layers).

Christie *et al*. (2015: 5-9) states that neural networks accommodate categorical inputs effortlessly compared to other machine learning techniques. The complexity of a neural network provides the opportunity to obtain a very strong and versatile predictive model.

Due to the illusion that neural networks are mysterious, and the current hype that is associated around the use of neural networks, the urge to disregard all other machine learning techniques other than neural networks exist. Neural networks, however, comes with some defects.

- Variable selection needs focus in modelling data with a neural network. Otherwise, overfitting is probable (Christie *et al*., 2015: 5-3).
- Objective function that is to be minimised may be non-convex, thus local minima enters the equation (Hastie *et al.,* 2008: 397).
- No clear rules exist to choose the number of hidden layers and hidden units of a neural network (Hastie *et al.,* 2008: 400).
- Careful consideration is necessary in choosing initial values for the weights with which the optimisation process starts with (Hastie *et al.,* 2008: 401)
- Difficult to interpret and explain the estimated weights and parameters of a neural network (Christie *et al*., 2015: 5-8).
- Outliers can pose a problem (Christie *et al*., 2015: 5-9).

Christie *et al*. (2015: 5-52) suggests conducting variable selection prior to training a neural network. Missing values are problematic during training of a neural network and during scoring observations by means of a neural network (Christie *et al*., 2015: 5-9).

3.6.5 How to avoid overfitting and local minima

In training a neural network caution concerning overfitting is important since neural networks often have numerous weights and as a result overfitting is a likely consequence (Hastie *et al*., 2008: 398). Therefore, the need to implement certain measures to avoid overfitting exists. For example, a limit on the number of iterations involved in the optimisation process of the weights can be set (Hastie *et al*., 2008: 398). Another approach is to incorporate a validation set in the training

process, training is terminated when the performance of the trained model starts to deteriorate on the validation set (Hastie *et al*., 2008: 398). Thirdly, an approach similar to the regularisation imposed on the estimated coefficients in ridge regression (see Chapter 3.7), can be exploited to prevent overfitting in training a neural network (Hastie *et al*., 2008: 398). The technique is referred to as *weight decay* and effectively entails adding a penalty to the objective function that is to be minimised (Hastie *et al*., 2008: 398). In other words, as Mitchell (1997: 111) explains, *weight decay* is the reduction in the numerical value assigned to each weight during each iteration of the training process.

As mentioned before, the neural network poses the risk of ending up trapped in a local minimum. Two of the strategies Mitchell (1997: 104-105) suggests to attempt to avoid this phenomenon:

- Implementation of stochastic gradient descent as an alternative to normal gradient descent.
- Use different starting weights to train several networks, choose the network that performs the best or use all the trained networks and obtain an average of all the output provided by the different networks.

Hastie *et al*. (2009: 401) warns against the option to use an average of the weights obtained by training several networks with different starting values as the final weights of the neural network. Hastie *et al*. (2009: 401) rather suggests training neural networks on different random samples of the observed training dataset and to average the resulting predictions. There exists similarity in this approach and the method that can lead to confidence intervals for neural networks by making use of bootstrapping.

In conclusion, the technique called *dropout* can also be utilised to prevent overfitting. In addition, *dropout* can aid as a mechanism to introduce noise in the training process and subsequently will lead to a model that will potentially generalise better to new data (Blanchard & Wells, 2018: 1-17). Blanchard and Wells (2018: 1-17) mentions that *Dropout* is implemented by randomly eliminating inputs and hidden units (or both the former and the latter) during the training process by multiplying the output of the component with zero. The random removal of components continues until convergence is reached or, alternatively, until the maximum number of iterations are reached (Blanchard & Wells 2018: 1-17).

SAS Enterprise Miner allows specification of preliminary training of the neural network to take place, during which the neural network is trained for a small number of iterations but for multiple randomly assigned initial weight values (*SAS Institute Inc*. 2018f). The most optimal estimated weights from this process will then be used as the initial weight values in the main training to follow and thus acts as calculated guesses (*SAS Institute Inc*. 2018f).

SAS Enterprise Miner allows a selection of an integer ranging from 1 to 64 as the number of hidden units with a default value of 3 (*SAS Institute Inc*. 2018e). A possible approach to decide how many hidden units to specify, is to incrementally increase the number of hidden units and compare the performance of each resulting neural network to the preceding neural networks (*SAS Institute Inc*. 2018f). Stop the process of addition of hidden units once it becomes evident that the performance is deteriorating (*SAS Institute Inc*. 2018f). Setting a limit on the maximum training time and the maximum number of iterations are possible (*SAS Institute Inc*. 2018e). Several optimisation techniques used to train the weights are available in SAS Enterprise miner of which back propagation is one (*SAS Institute Inc*. 2018e). SAS Enterprise Miner allows the following model selection criteria as part of the *stopped training* process described in Section 3.6.3: profit/loss, misclassification and average error (*SAS Institute Inc*. 2018e).

3.7 DESCRIBING REGULARISATION, THE BIAS – VARIANCE TRADE – OFF AND THE CURSE OF DIMENSIONALITY

In the Chapter 3 thus far, three common concepts in machine learning and statistical modelling have been mentioned namely, regularisation, the trade – off between bias and variance and the phenomenon called the *curse of dimensionality* (Bellman, 1961 cited in Hastie *et al.,* 2009: 22). Chapter 3.7 aims to briefly discuss these concepts.

Hastie *et al*. (2009: 57, 61) explains that regularisation, as opposed to best subset variable selection (determining the best subset of variables across the possible subsets), has the advantage that regularisation suffers less from an increase in variance compared to subset variable selection because regularisation is less of a discrete process. High variance refers to differing results from consecutive training of the same model on different samples. Variable selection, for example best subset selection, has a discrete nature in the sense that a variable is either included or not (Hastie *et al*. 2009: 61). While regularisation, by means of the introduction of a penalty term as part of the optimisation, shrinks the coefficients closer to zero rather than disregarding certain variables in a discrete (Yes/No) manner (Hastie *et al*. 2009: 61). With regards to regression two common regularisation techniques are Ridge regression and the Lasso. The form of the penalty term stand central in the difference between Ridge regression and the Lasso (Hastie *et al*. 2009: 68). Regularisation is not restricted to regression only but is also applicable to other techniques such as neural networks and SVM as mentioned in Chapter 3. It might occur by another name, for example *Weight Decay* (as mentioned in Section 3.6.5) is a form of regularisation in neural networks (Hastie *et al*. 2009: 63).

Hastie *et al*. (2009: 16) compares a linear model fitted by least squares and the k – nearest neighbours algorithm in terms of the bias − variance trade – off. This study also explains bias − variance trade – off by referring to a linear decision boundary fitted by least squares and a decision

boundary fitted by k – nearest neighbours without going into the detail behind the estimation process of each. Since the linear decision boundary fitted by least squares are restricted to attain a linear structure, irrespective of whether a nonlinear structure might be more appropriate, it potentially suffers from high bias (Hastie *et al*., 2009: 16). Whereas, the strict structural assumption does cause the decision boundary to be stable in the sense that the estimated parameters obtained on another sample dataset will tend not to differ by much (Hastie *et al*., 2009: 16). The opposite is true with k – nearest neighbours since no structural assumptions are made at all (thus low bias), the estimate of each point in the decision boundary rely solely on the neighbouring points which can cause the decision boundary to be wavy (Hastie *et al*., 2009: 16). The latter causes the decision boundary to differ significantly between different training samples and this leads the instability associated with the decision boundary (Hastie *et al*., 2009: 16).

Hastie *et al*. (2009: 25) present a visual appraisal of the *curse of dimensionality* by providing a plot of the MSE as an increasing function of the number of inputs p. Thus, a drastic increase in the number of inputs while the size of the training data remains constant, can result in an exponential increase of a measure such as MSE (Hastie *et al*., 2009: 24).

3.8 SUMMARY

Chapter 3 focussed on the description of four machine learning techniques, namely, logistic regression, decision trees, SVM and neural networks. Throughout the chapter, the contrast in complexity among the techniques is evident. The advantages and disadvantages of the techniques are highlighted. In addition, information on how the software of choice in this study, SAS Enterprise Miner, implements the techniques is discussed. An important feature of this chapter is the comparison between regression and neural networks. Chapter 4 focusses on the implementation of the techniques described in Chapter 3.

CHAPTER 4

APPLICATION OF ALGORITHMS

4.1 INTRODUCTION

The description of both the data (Chapter 2), as well as, the machine learning techniques (Chapter 3) paved the way for the discussion of Chapter 4. Chapter 4 focus on the application of each technique Chapter 3 described on the data Chapter 2 presented.

The performance of the different models is compared by considering fit statistics calculated on the same sets of training and validation data. Also, the training time of the different models receives consideration. Chapter 4 sequentially describes different tactics, each believed to influence the performance of the models, as well as possibly addressing the mentioned presence of class imbalance in the data. Among these tactics are manipulation of the input features, introduction of prior probabilities and construction of a decision matrix. Also, the addition of variables that have the potential to improve the model's ability to predict the response are described. In some instances, the variables require manipulation to ensure efficient participation in the modelling process. Chapter 4 thus contributes to the discussion around variable construction and data manipulation prior to modelling, that Chapter 2 described.

A series of champion models will be obtained after implementation of the different tactics to improve the model performance. The series of champion models are finally evaluated on a test data set, not involved in the training process. Details regarding the functionality of Enterprise Miner by means of output, as well as, workflow diagrams are presented in this chapter, as well as in Appendix A.

4.2 UNBALANCED DATA

It is often that the frequency of either of the two categories of the binary variable differs substantially (Duggal *et al*. 2016a: 472; Zheng *et al.*, 2015: 7112). This leads to what is called an imbalanced dataset (Chawla, 2010 cited in Zheng et al., 2015: 7113; Duggal *et al*. 2016a: 472). This phenomenon is not uncommon with response variables related to clinical outcomes such as whether a patient is diagnosed with a particular disease, for example cancer (Mazurowski, Habas, Zurada, Lo, Baker, Tourassi, 2008: 427). The classification ability of algorithms is affected negatively by the presence of class imbalance (Mazurowski *et al.*, 2008: 427). In the event of a low frequency for the outcome of interest, for example a readmission, a classifier calibrated on the minimisation of the MSE, may more often than not classify cases to not being a readmission (Mazurowski *et al.*, 2008: 429).

In terms of disease detection, if 5 percent of cases in a test dataset have cancer and the model classify all cases in the test dataset as not having cancer, it leads to a 95 percent correct classification, however, all the cases having the most important outcome, namely cancer, is classified incorrectly (Mazurowski *et al.*, 2008: 428). As mentioned, variables such as readmissions also suffer from class imbalance.

The literature proposes several techniques to apply in the event of class imbalance of the response variable. The simplest of these techniques are *undersampling* and *oversampling*. *Undersampling* consists of separately sampling the same number of events and non – events, the training data then include both samples, combined into one dataset (Mazurowski *et al.*, 2008: 430). Whereas, *oversampling* occurs by duplicating events in the training data in order for the number of events and non – events to match (Mazurowski *et al.*, 2008: 430).

In the applications of algorithms, the accuracy of the technique is (Mazurowski *et al.*, 2008: 428):

$$\frac{True\ positive(TP) + True\ negative(TN)}{True\ positive(TP) + True\ negative(TN) + False\ positive(FP) + False\ negative(FN)} \tag{4.1}$$

where in the context of readmissions

TP – Correctly classify a case as a readmission

TN – Correctly classify a case as not being a readmission

FP – Incorrectly classify a case as a readmission (the case is in fact not a readmission)

FN – Incorrectly classify a case as not being a readmission (the case is in fact a readmission)

Accuracy of a classifier depends on the decision threshold implemented by the classifier on the outcome variable of interest (Mazurowski *et al.*, 2008: 429). The Receiver Operator Characteristic (ROC) curve and the area under the ROC curve (AUC) is popular for model evaluation in clinical applications (Duggal *et al*., 2016b: 523; Obuchowski, 2003 cited in Mazurowski *et al.*, 2008: 429). Mazurowski *et al.* (2008: 429) points out that the ROC curve graphically displays sensitivity versus false positive rate at different decision thresholds, therefore the preference to ROC, especially in the light of class imbalance. Duggal *et al*. (2016b: 523) agrees that ROC is a good choice of metric for situations where the focus is mainly on predicting the event, which has a low occurrence, correctly. Mazurowski *et al*. (2008: 430) describes AUC as being independent to imbalance in the training data. Although not ideal, the majority of machine learning techniques train by optimising the overall accuracy (Longadge, Dongre & Malik, 2013: 2).

An AUC of one will indicate an impeccable classification model, whereas AUC of 0.5 will be equivalent to a coin toss (Futoma *et al*., 2015: 233). Although the wide use of AUC is critisised by

some, Tong *et al.* (2016: 2) agrees that it is an appropriate measure to use in the prospects of building a readmission model. It provides an indication of the model's discrimination ability, that is, the model's ability to distinguish between clusters of patients with either high or low risk (Tong *et al.*, 2016: 2). Whereas, the calibration ability of a model measures the ability of the model to obtain predictions that is close to the actual risk of readmission (Tong *et al.*, 2016: 2).

However, the necessity of confirmation of the model performance by means of an additional measure exists (Futoma *et al.*, 2015: 235). See Appendix A for illustrations of ROC curves.

Table 4.1: Description of performance measures

Name	**Description**	**Usage**
AUC (ROC Index)	Measures true positive rate versus false positive rate (Mazurowski *et al.*, 2008: 429)	Popular for imbalanced datasets (Duggal *et al.*, 2016a: 473)
Recall	True positive rate (sensitivity) (Billings *et al*., 2012: 3) $\frac{Number\ of\ patients\ correctly\ predicted\ as\ a\ readmission}{Number\ of\ patients\ actually\ readmitted}$	Popular when main goal is to correctly classify less frequently observed class (Duggal *et al.* 2016b: 523)
Specificity	True negative rate, (Billings *et al*., 2012: 3) $\frac{Number\ of\ patients\ correctly\ predicted\ as\ a\ non-readmission}{Actual\ number\ of\ patients\ not\ readmitted}$	Provide additional insight in the model performance.
Accuracy	Proportion of correctly classified cases (*Duggal et al*., 2016a: 473) $\frac{Number\ of\ patients\ classified\ correctly}{Total\ number\ of\ patients}$	Popular to obtain indication of global model performance (*Duggal et al*., 2016a: 473)

Positive predicted value (PPV) or precision	Given a particular threshold, the number of readmissions as a proportion to the number of identified high risk observations, (Billings *et al*., 2012: 3) $\frac{Number\ of\ \ patients\ correctly\ \ predicted\ to\ be\ readmitted}{Number\ of\ \ patients\ \ predicted\ to\ be\ readmitted}$	Commonly used if a high cost can result from false positive predictions (Duggal *et al*., 2016a: 473)

A consequence of under – sampling is the resulting loss of data therefore, the introduction of cost matrices follows as part of the training process to address the problem that imbalanced datasets pose, without disturbing the distribution of the response variable (Longadge *et al.*, 2013: 2). Therefore, this study introduces the following cost (profit) matrix,

$$\begin{array}{cc} & Classification \\ & \begin{array}{cc} 1 & 0 \end{array} \\ Actual \begin{array}{c} 1 \\ 0 \end{array} & \begin{bmatrix} 4y-x & -4y \\ -x & 0 \end{bmatrix} \end{array} \tag{4.2}$$

where x is the average cost for one accommodation day in hospital, per admission in general.

The total loss for a readmission classified as not being a readmission and consequently no interventions are applied on discharge of the index admission is, $4y$. This study estimates that a readmission will stay approximately 4 days in hospital, thus y boils down to the average cost per readmission per day.

The total cost associated with interventions should a patient be predicted as a potential readmission at discharge of the index admission is assumed to be x , where x is the average cost per hospital admission per day. The cost of one additional accommodation day is thus used as a proxy for the cost of the intervention, but it does not necessarily imply that the intervention consists of an additional day in hospital.

The total profit, should a patient be predicted as a high risk for readmission and under the assumption that the interventions successfully prevent this, is $4y-x$

This is quite a progressive cost matrix and severely penalises wrong classifications.

4.3 FEATURE MANIPULATION

Continuous variables such as age and Body Mass Index (BMI) can be included as either a continuous variable or discrete variable. Duggal *et al*. (2016b: 522) uses discretisation to transform the following variables to discrete variables: age and number of diagnoses and procedures.

The approach that this study follows is to apply the decision tree algorithm in SAS Enterprise Miner. Section 3.4.5 describes that the higher the result of the calculation of the *logworth*, equation (3.28), the more significant is the variable as a potential split point. The *logworth* of each variable's optimal splitting point is reported in the software's output. Each pair of variables, for example ICU days as an interval variable versus ICU days as an indicator variable is run through a decision tree with 30 – day readmission as binary response variable. Between the two formats of the variable, the one that has the highest associated *logworth* is used as input variable in the modelling process to follow in this chapter. Both the categorical age variables and the categorical BMI variables are available in the data via the application of a user defined SAS format or simple programming logic. The difference between these BMI and age categorical variables are the width of the respective intervals. Age variable 1 varies from 0 to greater than 100 years with intervals [0, <1], [1,14], [15,34], [35,54], [55,84], [85,>100]. Age variable 2 varies from 0 to greater than 100 years with intervals [0, <1], [1,4], [5,9], [10,14], [15,19], …., [80,84], [85,>100]. Mediclinic International customarily uses one of the following BMI classifications

BMI1:

- (0, 18.5) − Underweight
- (0,16) − Underweight
- [16,17) − Underweight
- [17,18.5) − Underweight
- [18.5,25) − Normal
- [25,30) − Overweight
- [30,35) − Obese
- [35,40) − Obese
- [40,>40) − Obese

BMI2:

- (0,16) − Severe Thinness
- [16,17) − Moderate Thinness
- [17,18.5) − Mild Thinness
- [18.5,25) − Normal

- [25,30) − Pre - Obese
- [30,35) − Obese Class I
- [35,40) − Obese class II
- [40,>40) − Obese class III

Table 4.2: *Logworth* per variation of input variable

Variable	**Structure**	***Logworth***	**Chosen**
Age	Continuous	101.5282	No
Age1	Ordinal	90.2921	No
Age2	Ordinal	102.5282	Yes
Admission time (hour)	Nominal	66.0374	Yes
Admission time (morning, afternoon, evening)	Nominal	62.3414	No
BMI	Continuous	7.1612	No
BMI1	Ordinal	64.1543	Yes
BMI2	Ordinal	63.8533	No
BMI indicators (Obese, severe obese, morbidly obese and underweight)	Binary	3.6927 (average)	No
Discharge time (hour)	Nominal	21.7184	Yes
Discharge time indicator (morning, afternoon, evening)	Nominal	9.6900	No
ICU days	Interval	20.0550	No

ICU days indicator	Binary	21.2011	Yes
High care days	Interval	19.4908	Yes
High care indicator	Binary	16.0599	No
Major theatre minutes	Interval	129.5189	No
Major theatre indicator	Binary	130.3178	Yes
Prosthesis cost	Interval	12.6398	Yes
Prosthesis indicator	Binary	1.9185	No

Shadmi *et al.* (2015: 285) states that the reduction of the number of variables presented to the training process of the readmission model can occur by means of decision trees, this includes both classification and regression trees. According to Tong *et al.* (2016: 2), the importance of variable selection, but also the inability of traditional methods such as forward and backward variable selection to handle the large number of independent variables available in modelling readmissions, is apparent

There exist three instances where either variable reduction, or reduction of the number of levels of a categorical variable, prior to the actual modelling, is deemed necessary in this study. Firstly, the construction of a comorbidity and complication list, as Chapter 2 described. In addition, decision trees are used to collapse the numerous (greater than 1000) levels of the clinical sub – group variable into sixteen levels. In order to perform this task, the decision tree is trained on all the observations with the 30 − day readmission indicator as response variable and only the clinical sub − group as input variable. The minimum split and leaf size are 100 observations and the subtree pruning, as Section 3.4.5 describes, is disabled. A categorical variable with numerous levels is generally problematic for most algorithms, except for a decision tree (Christie *et al*., 2015: 9-20).

In addition, due to the significant amount of pharmacy transactions per account, the number of indicator variables generated by the process described in Section 2.3.4.1, is reduced by performing a correlation analysis. The correlation of each pharmacy indicator variable with the response (30 – day readmission) is calculated. Ranked on the absolute value of the correlation between each variable and the response, the top 40 variables from a total of 160 are retained while the rest is discarded.

The prior variable selection techniques which Chapter 1 and Chapter 4 describe can fail to identify certain crucial variables as significant, possibly due to a lack in volume of occurrence in the data. Should this be the case and there exist a strong clinical motivation that a variable should be considered in determining the probability of readmission, this study believes that the addition of such variables is essential. This study deems four types of comorbidities, either not featuring at all, or not having a substantial enough presence, in the list which Table 2.3 reports, as essential. The comorbidities are diabetes, hypertension, hemophilia and hyperlipidemia. A single binary variable per comorbidity (diabetes, hypertension, hemophilia and hyperlipidemia) indicating the presence of at least one code on the respective accounts relevant to any of these additional comorbidities are added to the dataset. Thus, for example the hypertension indicator can be triggered by multiple relevant ICD – 10 codes, in addition to the single code, I10, in Table 2.3.

4.4 APPLICATION AND TRAINING OF ALGORITHMS

It is common practice to utilise a training and a validation dataset in the process of calibrating a model. This is an attempt to avoid overfitting, as is confirmed in Mazurowski *et al*. (2008: 430). It is important not to report the performance of the resulting model based on the validation dataset as a measure of the model's final performance. Since the validation dataset plays an integral role in the training of the model, together with the training dataset. It is therefore advisable to report the performance of the final model by means of a test dataset, which were not involved in the training process. Mazurowski *et al*. (2008: 430) points out the necessity that in both the training and validation dataset the frequency of the different classes that the outcome can attain, as well as the size of the respective datasets, must be similar.

4.4.1 Model characteristics

As mentioned, four types of machine learning algorithms are considered in this study. Chapter 3 provided an explanation of the underlying theory of each technique. It is possible to implement different variations of each modelling technique. The variations differ in regards with, for example, the involved hyperparameters, maximum allowed iterations and presence or absence of variable selection prior or during the training of the model. A combination of models is also considered as a candidate model. This is usually referred to as an ensemble model.

Figure 4.1 illustrates the workflow diagram constructed in SAS Enterprise Miner. Each component is numbered for ease of reference.

4.4.1.1 Data source

The diagram starts with a node that reads the dataset from the relevant library. This node enables the user to set the properties of each variable. The target and input variables are specified, as well as the type of each variable, for example binary, ordinal or interval. This node has the

functionality to specify the prior probabilities, as well as the cost matrix that assist in the training of the algorithms.

4.4.1.2 Data partition

The data partition node separates the data randomly into a training and a validation dataset. This node provides the opportunity to specify the proportion of observations to be allocated to the training and validation datasets respectively. This study opts for an equal split of the observations between the training and validation datasets. Each subsequent modelling node will make use of the training and validation datasets in conjunction with each other in order to obtain an estimated model that has good generalisation capabilities. It is crucial to keep in mind that the performance measures of the model on both the training and validation datasets are biased. The bias is ascribed to the fact that both datasets are involved in the model estimation. To obtain an unbiased performance measures of the trained model, a test dataset, not involved in the training at all, is needed. The temptation to adjust the model if the performance on the test dataset is not satisfactory should to be avoided. If any changes are made to the model after evaluation on the test dataset, a new test dataset is necessary.

4.4.1.3 Decision tree

Chapter 3.4 provides a detailed discussion regarding the decision tree algorithm that SAS Enterprise Miner implements. Chapter 3.4 mentions at least one disadvantage of decision trees, namely, instability. To possibly counter the instability of decision trees, as well as, due to the pursuit of a decision tree without leaves having only a couple of observations, the minimum leaf size and split size is set to be 100 observations. The leaf size restriction prohibits a node to split, if either of the resulting leaves (or nodes, depending on the position in the tree) will have less than 100 observations. The split size property ensure that the algorithm avoids splitting nodes with less than 100 observations. Regarding the subtree pruning described in Section 3.4.5, the pruning can be disabled and therefore the tree can grow as large as the number of significant variables, as well as the minimum leaf and split size allows it to grow. In addition, the default status quo is that prior probabilities and cost matrices do not affect whether a variable is split upon or not, in other words the growth of the tree (*SAS Institute Inc*. 2018c). In effect, this implies that the prior probabilities and decision matrices are not involved in the parameter estimation of the decision tree. However, this default behaviour can be adjusted for the prior probabilities and cost matrices to participate in the split search algorithm that SAS Enterprise Miner implements. Contrasting to tree growth, the prior probabilities will automatically participate in the pruning of trees if the subtree pruning property is enabled (*SAS Institute Inc*. 2018c).

4.4.1.4 Regression

Since the response variable is binary in this study, logistic regression is appropriate. The default link function for the regression node, namely the logit function, is used. The well − known forward variable selection technique is chosen, rather than the stepwise variable selection technique, in order to save on training time. In this study the training criterion is either validation misclassification or average validation profit.

4.4.1.5 Neural network

The hyperbolic tangent hidden layer activation function is used. The network is a one − layer network, with three hidden units. Stopped training, as Section 3.6.3 describes, with the training criterion of either average error or average profit is implemented. The maximum number of iterations are set to be 50 while the training time is restricted to four hours. Although possible, explicit specification of a weight learning technique for example back propagation, is not necessary, since the algorithm decides on an appropriate weight learning technique based on the number of weights to be estimated. Among the possible training techniques is back propagation as Chapter 3 describes.

4.4.1.6 SVM

The maximum number of iterations that the SVM algorithm has to its availability to produce a reasonable classifier is set to 25.

4.4.1.7 Variable selection trees

Seven decision trees are fitted on a random sample of observations from the training data. The leaf size and split size of each tree is set to five observations. This property provides an opportunity for all variables to be in contention to be a splitting variable. Thus, a binary variable with a low number of either positive (indicated by 1) or negative (indicate by 0) responses across all observations are not prohibited to be a splitting variable (while it might be highly significant) due to the leaf and split size restriction. The number of surrogate rules is set to be one. Christie *et al.* (2015: 9-18) recommends this specification in the event of the decision tree acting as a variable selection technique rather than as a model that is tasked to provide predictions. Christie *et al.* (2015: 9-18) base this recommendation on the fact that, setting the number of surrogate rules to one rather than keeping it fixed at the default of zero, allows variables to be deemed significant predictors despite being statistically related, and thus possibly redundant, to variables split upon earlier in the tree. Thus, all variables are presented a fair opportunity to act as a splitting variable in any of the decision trees and consequently have the opportunity to be involved in the training of models in the subsequent nodes.

This study attempts to mitigate the possible increase in instability that may occur because of the decrease in split and leaf size, by using multiple trees together with an appropriate combination rule in the Metadata node (Section 4.4.1.8). This chapter introduces the variable selection by means of the decision trees, in order to allow an increase in complexity of the hyperparameters of the subsequent nodes due to the reduction in input variables. For example, due to the reduction in the number of input features, an increase in the number of hidden units and/or iterations of the subsequent modelling nodes, is possible. Thus, the modelling nodes following the metadata node are more complicated as its counterparts in Section 4.4.1.3, Section 4.4.1.4, Section 4.4.1.5 and Section 4.4.1.6 but due to the preceding variable selection, have lesser of chance to over fit or to result in the computer running out of memory.

4.4.1.8 Metadata

The metadata node combines the results of various variable selection techniques and passes only the selection of variables on to subsequent modelling nodes. The rule can be set to be *any* (variable rejected by at least one decision tree is not available further on), *majority* (variables rejected by most decision trees are not available in the nodes to follow) or *all* (only if a variable is rejected by all the decision trees will the variable be ignored further on). This study make use of the *all* combination rule. The reason is that the instability of the decision trees together with the *any* option can result in the unavailability of most of the variables for the modelling nodes following the metadata node, should the majority of variables be rejected in at least one decision tree. The *all* option provides a more reliable result since seven decision trees finding a variable insignificant do provide satisfactory evidence that the variable is truly insignificant.

4.4.1.9 Neural network

This neural network differs from the neural network in Section 4.4.1.5 in terms of an increase in the upper limit on the number of allowed iterations to 300. Secondly, the number of hidden units are set to ten instead of three.

4.4.1.10 SVM

The only difference in comparison with the SVM of Section 4.4.1.6 is an increase in the maximum number of iterations to 150.

4.4.1.11 Regression

This regression model trains by implementation of the stepwise variable selection technique and, like the regression model of Section 4.4.1.4 the selection criterion is either validation misclassification or validation profit.

4.4.1.12 Neural network

The neural network is increased in complexity by allowing at most 300 iterations and six hidden units.

4.4.1.13 Ensemble

The ensemble node combines the predictions of all the models connected to it. The possible classification rules for a binary response are the average posterior probability of all models, maximum posterior probability across all models or a majority vote to determine the predicted class of the observation. This study considers both the average posterior probability and maximum posterior probability rule. Node 13 implements the maximum rule.

4.4.1.14 Ensemble

Node 14 implements the average rule as mentioned in Section 4.4.1.13.

4.4.1.15 Model comparison

The model comparison node is useful to compare all the models appearing in the workflow diagram with each other, based on a specified prediction measure. As mentioned before, the ROC index (area under the ROC curve) is a versatile measure and the proportion of the primary outcome does not affect it as severely. The model comparison node identifies the champion model based on the specified metric (ROC index) but also provides a summary of other fit statistics of each model.

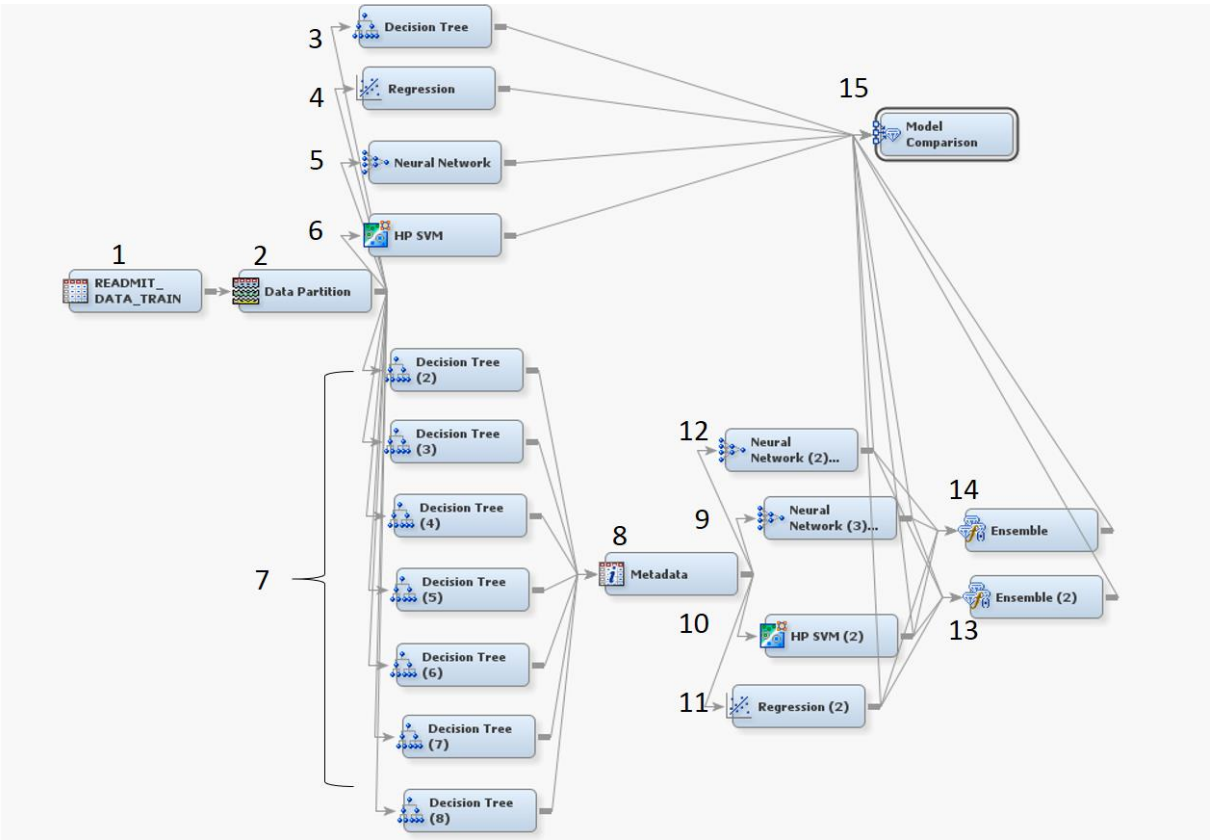

Figure 4.1: SAS Enterprise Miner workflow diagram

4.5.1 Training on unbalanced data

The data suffers from severe class imbalance. As mentioned, the tendency to classify observations into the common secondary class, namely non – readmissions, will occur often due to the class imbalance. In the case of considering the complete unbalanced dataset, it increases the required computing resources and will potentially lead to an increase in training time. Combining a large dataset with a computing expensive technique in its own right, namely the SVM, lead to difficulties regarding node 6 and consequently the SVM is unable to complete its training in Section 4.5.2. As mentioned in Section 3.5.4, the training process of the SVM is resource intensive due to the need to obtain the inverse of a data matrix, which can involve intense computation time, especially for enormous datasets (Marsland 2009: 119). However, after a process of variable selection by means of the decision trees as Section 4.4.1.7 described, the SVM successfully completes its training.

4.5.2 Training on unbalanced data with inclusion of cost matrix

In Section 4.5.2, the modelling occurs on the unbalanced data with the inclusion of the cost matrix of Chapter 4.2.

SAS Institute Inc. (2018b) confirms, as mentioned before, in the case of unbalanced data (say scarce primary outcome) the algorithm tends to classify cases to the secondary class (not a readmission). *SAS Institute Inc*. (2018b) points out that introducing a cost matrix with a substantial profit assigned to correctly classifying the rare event opposed to correctly classifying the common event, will force the algorithm not to fail to classify cases to the primary class as frequently as before.

The long training time of especially the regression and neural network nodes is evident in Table 4.3. The measures that Table 4.3 reports are ROC, MSE, average profit and T (time) in minutes. Concerning the measure used for model comparison, namely ROC index, the regression model of node 4 performs the best with a satisfactory ROC of 0.774 on the validation data. In terms of validation profit, which was involved in the training of the models (not the parameter estimation, see Section 4.5.3), the regression model of node 4 also outperforms the other models. The SVM of node 10 performs poorly while the SVM of node 6 prematurely stopped training due to memory limitations.

Table 4.3: Training models on unbalanced data with inclusion of cost matrix

	Training			**Validation**			
Model	**ROC**	**MSE**	**Profit**	**ROC**	**MSE**	**Profit**	**T**
Regression Node 4	0.775	0.0884	-1170.43	0.774	0.0885	-1181.75	139
Neural Net Node 9	0.772	0.0886	-1208.9	0.771	0.0887	-1208.81	153
Ensemble Node 14	0.772	0.0898	-1337.68	0.771	0.0898	-1337.42	8
Ensemble Node 13	0.772	0.0910	-1207.39	0.771	0.0910	-1210.58	8
Regression: Node 11	0.771	0.0888	-1233.24	0.77	0.0888	-1239.67	26

Neural Net Node 12	0.77	0.0888	-1224.33	0.77	0.0888	-1223.51	111
Neural Net Node 5	0.769	0.0899	-1204.17	0.769	0.0899	-1205.17	104
Decision Tree Node 3	0.76	0.0899	-1191.33	0.759	0.0899	-1196.65	13
SVM Node 10	0.705	0.1126	-3029.36	0.706	0.1126	-3031.43	9

From Table 4.4, the high specificity can be ascribed to the imbalance in the dataset, despite the efforts to adjust for this by means of introduction of a cost matrix. The high accuracy is somewhat misleading due to the relatively low recall. The high accuracy reflects mainly the high specificity. The poor performance of the SVM is once again evident. The advantage of combining models by means of maximum posterior probability is clear from the high recall of node 13, however, the average of the posterior probabilities has the opposite effect.

Table 4.4: Classification table based on unbalanced data with specification of cost matrix

Node	Role	FN	TN	FP	TP	Recall	Specificity	Accuracy
Node 3	TRAIN	84425	739036	14243	14827	14.94%	98.11%	88.43%
Node 3	VALIDATE	84373	738893	14387	14881	14.99%	98.09%	88.42%
Node 4	TRAIN	87268	745560	7719	11984	12.07%	98.98%	88.86%
Node 4	VALIDATE	87219	745616	7664	12035	12.13%	98.98%	88.87%
Node 5	TRAIN	95423	750797	2482	3829	3.86%	99.67%	88.52%
Node 5	VALIDATE	95384	750746	2534	3870	3.90%	99.66%	88.51%
Node 12	TRAIN	89095	747681	5598	10157	10.23%	99.26%	88.89%
Node 12	VALIDATE	89088	747725	5555	10166	10.24%	99.26%	88.90%
Node 9	TRAIN	88595	747500	5779	10657	10.74%	99.23%	88.93%
Node 9	VALIDATE	88638	747597	5683	10616	10.70%	99.25%	88.94%
Node 10	TRAIN	92042	748469	4810	7210	7.26%	99.36%	88.64%
Node 10	VALIDATE	92088	748544	4736	7166	7.22%	99.37%	88.64%
Node 11	TRAIN	87436	745833	7446	11816	11.91%	99.01%	88.87%
Node 11	VALIDATE	87508	745899	7381	11746	11.83%	99.02%	88.87%
Node 14	TRAIN	91551	748597	4682	7701	7.76%	99.38%	88.71%
Node 14	VALIDATE	91588	748662	4618	7666	7.72%	99.39%	88.72%
Node 13	TRAIN	82998	742180	11099	16254	16.38%	98.53%	88.96%
Node 13	VALIDATE	83009	742231	11049	16245	16.37%	98.53%	88.97%

4.5.3 Training on balanced (undersampled) data with prior probabilities

In Section 4.5.3, as *SAS Institute Inc*. (2018b) suggests, models are trained on a dataset originating by randomly sampling an equal number of events and non − events form the population. This should occur in conjunction with the specification of the correct prior probabilities based on the population. Christie *et al*. (2015: 6-20) warns against a possible misconception of the model's prediction ability concerning the population data, if the training process fail to adjust model predictions for the population prior probabilities. Christie *et al.* (2015: 6-20) mentions that undersampling, without adjusting for it, results in predictions based on the training sample, which does not reflect the occurrence rate of the event in the population.

SAS Institute Inc. (2018c) remarks that specification of prior probabilities does not influence parameter estimation. However, specification of prior probabilities does affect the estimates provided by the model and thus affect the misclassification rate, profit and loss, as well as, certain other common fit statistics (*SAS Institute Inc*. 2018c). Undersampling together with a failure in specifying appropriate prior probabilities causes the model to generate over estimated posterior probabilities for the occurrence of the scarce event (*SAS Institute Inc*. 2018c). Decreasing the prior probabilities to match the population proportion of the scarce event will decrease the estimated posterior probabilities (*SAS Institute Inc*. 2018c). As a result, specification of prior probabilities will influence the model selection (model selection in regression, pruning of splits in decision trees and the stopped training in neural networks) in the modelling nodes (*SAS Institute Inc*. 2018c).

Table 4.5 reports the fit statistics for 10 models trained on a balanced data set. The prior population probabilities are specified as part of the training process, but no cost matrix is presented to the algorithms. Since the algorithms do not have a decision matrix to its availability, misclassification and average squared error on the validation dataset is used as training measures. From Table 4.5 it is clear that in terms of the ROC index, the logistic regression model (Node 4 in Figure 4.1) and the Neural network (Node 12) are joint best. The logistic regression model slightly outperforms the neural network in all the remaining measures. The misclassification rate (MCR) is also reported in Table 4.5. Both neural networks with preceding variable selection performs better in comparison with the neural network without preceding variable selection.

Although, these two neural networks have both more hidden units than the neural network of Node 5, there exists a substantial decrease in training time due to the variable reduction. Overall, training models on a dataset originating from the process of undersampling results in a substantial decrease in the total training time per modelling node, due to the reduction of the size of the data. If the data consist of several nominal variables, each with multiple levels, a large dataset can pose problems in training a computing expensive technique such as the SVM. The results of the SVM provides evidence that it is important to evaluate model performance by considering multiple fit

statistics, since with regards to ROC the SVM has reasonable results but considering the misclassification rate, the SVM performs no better than a coin toss. The reduction in training times is apparent compared to Table 4.3.

Although Figure A.2 in Appendix A does not provide a clear illustration of the champion model, it is evident that the blue and pink line tend to be below the other lines, these two lines correspond to the two SVM models and provides further evidence of unsatisfactory prediction performance.

Table 4.5: Training models on balanced data with specification of prior probabilities

	Training			**Validation**			
Model	**ROC**	**MSE**	**MCR**	**ROC**	**MSE**	**MCR**	**T**
Regression Node 4	0.776	0.1926	0.4480	0.775	0.1931	0.4481	17
Neural Net Node 12	0.776	0.1927	0.4537	0.775	0.1932	0.4530	10
Decision Tree Node 3	0.773	0.1934	0.4339	0.772	0.1938	0.4338	2
Ensemble Node 14	0.773	0.3358	0.4794	0.772	0.3356	0.4794	2
Ensemble Node 13	0.772	0.3033	0.4357	0.771	0.3031	0.4355	2
Regression Node 11	0.771	0.1947	0.4521	0.77	0.1950	0.4521	2
Neural Net Node 9	0.771	0.1948	0.4625	0.769	0.1953	0.4623	8
Neural Net Node 5	0.768	0.1964	0.4714	0.767	0.1967	0.4712	16
SVM Node 6	0.767	0.2126	0.5000	0.765	0.2129	0.5000	3

SVM Node 10	0.759	0.2407	0.5000	0.758	0.2407	0.5000	2

There is a reduction in the accuracy of the models in Table 4.6 in comparison to the models in Table 4.4. This should be viewed in the light of a change in the distribution of the response variable due to the undersampling. It is important to note that fit statistics such as MCR and MSE is not altered to conform with the specified prior probabilities since these measures' main objective is to provide insight in the model performance on the training data as is (*SAS Institute Inc*. 2018c). In terms of recall, the decision tree of node 3 and the regression model of node 4 performs the best. However, the recall is in general quite low due to the substantial amount of FN cases. Exact conclusions of the model performance on population level cannot be drawn on fit statistics calculated on an undersampled dataset, except for the profit/loss fit statistic (*SAS Institute Inc*. 2018c).

Table 4.6: Classification table based on balanced data with specification of prior probabilities

Model Description	Role	FN	TN	FP	TP	Recall	Specificity	Accuracy
Node 3	TRAIN	85133	98252	1001	14119	14.23%	98.99%	56.61%
Node 3	VALIDATE	85098	98230	1023	14156	14.26%	98.97%	56.62%
Node 4	TRAIN	87809	98122	1131	11443	11.53%	98.86%	55.20%
Node 4	VALIDATE	87768	98071	1182	11486	11.57%	98.81%	55.19%
Node 5	TRAIN	92879	98566	687	6373	6.42%	99.31%	52.86%
Node 5	VALIDATE	92810	98536	717	6444	6.49%	99.28%	52.88%
Node 6	TRAIN	99246	99253	0	6	0.01%	100.00%	50.00%
Node 6	VALIDATE	99246	99252	1	8	0.01%	100.00%	50.00%
Node 12	TRAIN	89263	98449	804	9989	10.06%	99.19%	54.63%
Node 12	VALIDATE	89114	98443	810	10140	10.22%	99.18%	54.70%
Node 9	TRAIN	91041	98490	763	8211	8.27%	99.23%	53.75%
Node 9	VALIDATE	91014	98507	746	8240	8.30%	99.25%	53.77%
Node 10	TRAIN	99249	99253	0	3	0.00%	100.00%	50.00%
Node 10	VALIDATE	99251	99253	0	3	0.00%	100.00%	50.00%
Node 11	TRAIN	88695	98212	1041	10557	10.64%	98.95%	54.79%
Node 11	VALIDATE	88693	98208	1045	10561	10.64%	98.95%	54.79%
Node 14	TRAIN	94860	98955	298	4392	4.43%	99.70%	52.06%
Node 14	VALIDATE	94895	98977	276	4359	4.39%	99.72%	52.06%
Node 13	TRAIN	85062	97833	1420	14190	14.30%	98.57%	56.43%
Node 13	VALIDATE	84992	97790	1463	14262	14.37%	98.53%	56.45%

4.5.4 Training on balanced (undersampled) data with cost matrix as well as appropriate prior probabilities

The average profit and loss are adjusted for the prior probabilities and is thus preferred to be used to compare models with each other when modelling on underdamped data (*SAS Institute Inc*. 2018c). During the computation of the average profit based on the cost matrix, the prior probabilities specified are used to adjust the profit estimations (*SAS Institute Inc*. 2018c). Therefore, *SAS Institute Inc*. (2018b) advises to use both prior probabilities and a decision cost matrix if modelling is performed on an undersampled dataset.

Table 4.7 reports the fit statistics for 10 models trained on a balanced data set. The prior population probabilities are specified as part of the training process, as well as, the cost matrix is presented to the algorithms. Since the algorithms do have a decision matrix to its availability, average profit on the validation dataset is used as a training measure. From Table 4.7 it is clear that in terms of the ROC index, the logistic regression model (Node 4 in Figure 4.1) once again outperforms the other models considering most measures calculated on the training and validation dataset. Also, the Neural network (Node 12) are joint best in terms of ROC but fails to beat the logistic regression model of node 4 with regards to MSE and MCR. Once again, the ROC curves do not clearly distinguish between the respective model's performances considering Figure A.3.

Overall the measures in Table 4.5 (no cost matrix) and Table 4.7 (cost matrix) is similar however the additional measure of profit adds insight into the performance of the models and therefore this study suggest the inclusion of cost matrices in modelling. As mentioned, the cost matrix that the algorithms utilise is unrelenting towards misclassification and therefore a negative profit (loss) is produced. However, the maximisation of the profit, or as it ended up being, the minimisation of the loss, still provide valuable insight in the model performance.

Table 4.7: Training models on balanced data with specification of prior probabilities as well as cost matrix

	Training				**Validation**				
Model	**ROC**	**MSE**	**MCR**	**Profit**	**ROC**	**MSE**	**MCR**	**Profit**	**T**
Regression: Node 4	0.776	0.1927	0.4483	-1166.08	0.775	0.1932	0.4484	-1184.23	17
Neural Net: Node 12	0.776	0.1928	0.4553	-1168.65	0.775	0.1932	0.4546	-1173.26	10

Decision: Tree Node 3	0.773	0.1934	0.4339	-1175.18	0.772	0.1938	0.4338	-1180.60	2
Ensemble: Node 14	0.773	0.3370	0.4800	-1243.29	0.772	0.3368	0.4799	-1243.05	2
Ensemble: Node 13	0.772	0.3047	0.4370	-1265.96	0.771	0.3045	0.4370	-1286.68	2
Regression: Node 11	0.771	0.1947	0.4519	-1237.03	0.770	0.1950	0.4522	-1238.74	2
Neural Net: Node 9	0.771	0.1948	0.4647	-1227.92	0.769	0.1954	0.4644	-1230.08	8
Neural Net: Node 5	0.768	0.1964	0.4714	-1201.16	0.767	0.1967	0.4712	-1202.34	16
SVM: Node 6	0.767	0.2126	0.5000	-1344.49	0.765	0.2129	0.5000	-1360.81	4
SVM: Node 10	0.759	0.2407	0.5000	-1437.04	0.758	0.2407	0.5000	-1448.04	2

Table 4.8 reports similar result to Table 4.6. As expected, the addition of a cost matrix did not change the result obtained for the decision tree (see Section 4.4.1.3).

Table 4.8: Classification table based on balanced data with specification of prior probabilities as well as cost matrix

Model Description	Role	FN	TN	FP	TP	Recall	Specificity	Accuracy
Node 3	TRAIN	85133	98252	1001	14119	14.23%	98.99%	56.61%
Node 3	VALIDATE	85098	98230	1023	14156	14.26%	98.97%	56.62%
Node 4	TRAIN	87861	98127	1126	11391	11.48%	98.87%	55.17%
Node 4	VALIDATE	87827	98070	1183	11427	11.51%	98.81%	55.16%
Node 5	TRAIN	92879	98566	687	6373	6.42%	99.31%	52.86%
Node 5	VALIDATE	92810	98536	717	6444	6.49%	99.28%	52.88%
Node 6	TRAIN	99246	99253	0	6	0.01%	100.00%	50.00%
Node 6	VALIDATE	99246	99252	1	8	0.01%	100.00%	50.00%
Node 12	TRAIN	89622	98489	764	9630	9.70%	99.23%	54.47%
Node 12	VALIDATE	89471	98478	775	9783	9.86%	99.22%	54.54%

Node 9	TRAIN	91513	98527	726	7739	7.80%	99.27%	53.53%
Node 9	VALIDATE	91494	98570	683	7760	7.82%	99.31%	53.56%
Node 10	TRAIN	99249	99253	0	3	0.00%	100.00%	50.00%
Node 10	VALIDATE	99251	99253	0	3	0.00%	100.00%	50.00%
Node 11	TRAIN	88665	98206	1047	10587	10.67%	98.95%	54.81%
Node 11	VALIDATE	88711	98203	1050	10543	10.62%	98.94%	54.78%
Node 13	TRAIN	85358	97865	1388	13894	14.00%	98.60%	56.30%
Node 13	VALIDATE	85314	97825	1428	13940	14.04%	98.56%	56.30%
Node 14	TRAIN	94989	98969	284	4263	4.30%	99.71%	52.00%
Node 14	VALIDATE	95001	98989	264	4253	4.28%	99.73%	52.01%

Table 4.9 provides an indication of the variables that the variable selection decision trees in node 7 of Section 4.5.4 find significant. An argument that the variables “CODE_I10” and “HYPERTENSION” are correlated has merit. However, the variable “CODE_I10” is triggered by only one ICD code, whereas, the variable “Hypertension” is triggered by various codes, of which ICD code I10 is one. For most of the variables for example age and BMI, the clinical motivation for its significance is intuitive. It is interesting to notice that discharge hour is significant, and a possible scenario is that if the patient is discharged close to the change in personnel shifts that the discharge procedure such as medication reconciliation may be neglected, and this may lead to readmission. Also, it is noteworthy that the ATC classification that ended up being significant is N05AL01, Sulpiride. The variable resulting from the collapse of the variable containing the numerous clinical sub – groups into a nominal variable with only sixteen levels (Chapter 4.3) is also found significant.

Table 4.9: Significant variables in predicting readmissions

NAME	Description	LEVEL
ACCOM_DAYS	Accommodation Days	INTERVAL
AGE_GROUP2	Age	ORDINAL
ARR_METHOD	Arrival Method	NOMINAL
BMI_CLASS1	BMI	ORDINAL
CATH_IND	Catheter Indicator	BINARY
CODE_B95_3	Strep Pneumoniae as cause of dis classif other chapters	BINARY
CODE_I10	Essential primary hypertension	BINARY
DIS_HOUR	Discharge Hour	NOMINAL
EMER_ELEC	Emergency/Elective admission Indicator	NOMINAL
HC_DAYS	High Care Days	INTERVAL
HYPERLIPIDAEMIA	Hyperlipidaemia	BINARY
HYPERTENSION	Hypertension	BINARY
MED_SURG	Medical Surgical Indicator	NOMINAL
MJR_THT_IND	Major theatre minutes Indicator	NOMINAL
N05AL01	Sulpiride	BINARY
PRS_AMT	Prothesis Amount	INTERVAL

STAT_CLASS	Patient Type	NOMINAL
TOT_MINS	Theatre minutes	INTERVAL
SGRP_VAR_V2	Clincal sub - group	NOMINAL

4.6 SUMMARY

Chapter 4 sheds light on the modelling process in SAS Enterprise miner. This includes descriptions of the properties of the models that is trained in SAS Enterprise Miner, as well as, performance measures on the training and validation data. An explicit explanation of the different approaches regarding modelling is provided, namely, training on a population equivalent dataset, thus ignoring the imbalance in the data, in contrast with, modelling on a balanced dataset resulting from randomly sampling an equal proportion of events and non – events from the population. The technique, known as undersampling, has likely advantages but also needs to be utilised with the necessary care, otherwise distorted conclusion will be at the order of the day.

However, as mentioned model performance measured on training and validation data is biased since the training and validation data is used to make changes to the models on the fly. Thus, in order to obtain an unbiased indication of the models described in this chapter, Chapter 5 implements the respective champion models of Section 4.5.2 – Section 4.5.4 on a test dataset to which the models have no prior exposure. The test data resembles the population distribution of the response variable.

CHAPTER 5

DISCUSSION OF RESULTS AND IMPLEMENTATION

5.1 INTRODUCTION

Chapter 5 aims to conclude the discussion that this study presents by comparing the performance of the models of Chapter 4 on a test dataset, as well as, briefly reflect on the results of this study. Chapter 5 also identifies relevant challenges and shortcomings. In addition, future research and strategies for improvement is discussed. Considerations concerning implementation of a readmission model is also part of the scope of Chapter 5.

In order to obtain an unbiased indication of the models described in Chapter 4, Chapter 5 implements the respective champion models of Section 4.5.2 – Section 4.5.4 on a test dataset to which the models have no prior exposure. The test data resembles the population distribution in terms of the response variable. Additionally, Chapter 5 provides the opportunity to obtain an indication of the respective model performance in terms of measures that previously was not adjusted for the prior probabilities and, consequently, only resembles the model performance on the distorted training sample.

5.2 EVALUATION OF MODEL PERFORMANCE ON TEST DATA

Table 5.1 reports the performance measures per model when tasking the model to make a decision with the occurrence rate of readmissions as decision threshold. Suppose the occurrence rate (prior probability) of readmissions in the population is $p<1$. This implies that any patient admitted to hospital has a probability of p to be readmitted, irrespective of the patient's particular characteristics relative to the variables listed in Table 4.9.

If the characteristics in Table 4.9 are considered and the probability of readmission shifts to $\eta > p$ then the particular patient has an above average probability of readmission based on the patient's characteristics. Such a patient is then classified as a high − risk patient in terms of the outcome of readmission and thus the predicted value of the response variable will be positive, i.e. $\hat{y}=1$.

Conversely, if the probability of readmission after taking into account the characteristics of the patient, shifts to $\gamma < p$ then the patient has a below average risk to readmission and is considered as a low – risk case. Table 5.1 describes the model performance based on this reasoning.

Table 5.1: Performance measures on test data in terms of decision

Model	Recall Decision	Specificity Decision	PPV Decision	Misclassification Decision	Accuracy Decision
4.5.2	0.68917	0.74060	0.26225	0.26547	0.73453
4.5.3	0.68860	0.74176	0.26295	0.26452	0.73548
4.5.4	0.68865	0.74186	0.26305	0.26442	0.73558

If the customarily threshold of 0.5 is used instead of p to predict the response variable, it implies that the characteristics of the patient needs to be of such extreme nature to inflate the probability by a factor $\frac{0.5}{\eta}$. This is a stricter condition in order to predict a positive event ($\hat{y}=1$). As expected fewer positive cases will be predicted, leading to less FP cases, leading to an increase in PPV,

$$\frac{TP}{TP+FP\downarrow}=PPV\uparrow$$

As well as a decrease in Recall,

$$\frac{TP}{TP+FN\uparrow}=\text{Recall}\downarrow$$

Lastly, an increase in specificity

$$\frac{TN}{TN+FP\downarrow}=\text{specificity}\uparrow$$

Table 5.2: Performance measures on test data in terms of prediction

Model	Recall Prediction	Specificity Prediction	PPV Prediction
4.5.2	0.12179	0.99130	0.65192
4.5.3	0.11897	0.98966	0.60631
4.5.4	0.11838	0.98969	0.60578

5.3 IMPLEMENTATION

Billings *et al.* (2012: 3) considers the calculation of the mean resulting expenditure of patients readmitted, belonging to certain risk intervals, with the risk intervals defined based on the predictions provided by the model. Furthermore, estimations of the amount that can be allocated to implementation of interventions can be made by considering the potential cost saving if a certain predefined percentage of patients classified as high – risk for readmission, are assumed not to be readmitted due to the implemented interventions (Billings *et al.*, 2012: 3). The results of Chapter 5 assist in comparing the models. Implementation of the optimal model can contribute in the following way:

- Scoring hospital admissions with probabilities as data enter the data warehouse
- Risk adjust readmission rates by utilising the predicted probabilities of readmission. Perform benchmarking by calculating the ratio of observed to expected rates per grouping variable, for example per hospital,

$$\text{observed} = \text{Number of readmissions} \tag{5.1}$$

$$\begin{aligned}\text{Expected} &= \sum_{i=1}^{n}\left((0\times prob(y_i=0))+(1\times prob(y_i=1))\right)\\ &= \sum_{i=1}^{n}\left(0\times(1-prob(y_i=1))+(1\times prob(y_i=1))\right)\end{aligned} \tag{5.2}$$

$$= \sum_{i=1}^{n} prob(y_i=1) \tag{5.3}$$

  where $y_i = 1$ indicates a readmission

  $p(y_i = 1)$ is provided by the model

  n is the number of patients per level of the grouping variable, for example, the number of patients admitted per hospital.

- Develop dashboards in software that has the functionality to be accessible via mobile devices. The dashboards can be fed with data from the data warehouse, as well as, online capturing systems which the clinical workers can use to capture information as the information comes available. The dashboards display the calculated readmission risk for all current in – patients in each ward. This will enable clinical workers on duty to view the dashboard and constantly monitor the readmission risk of each patient. Certain variables will not be available or complete throughout the visit to hospital, for example, all the coding history will not be complete after the first day, as well as, the total accommodation days for the current visit to hospital will still be accumulating. Once the doctor gives the green light for discharge, all variables should be available and complete. A final readmission risk score is obtained via the model that runs in the background of the dashboard. Based on this score the clinical workers can implement predefined intervention measures that is

assigned to the particular risk score. Ideally, as mentioned, the total expected cost of the interventions should be proportional to the magnitude of the probability of readmission.

5.4 SHORTCOMINGS

The study experiences two main difficulties, namely, the unavailability of information that electronic health records (EHR) will provide. This study had to rely on coding information that is captured on an administration system, but the communication between the doctor and the clinical coder rely on a paper − based system. It is possible that an incomplete diagnostic statement provided by the doctor can result in incomplete coding information. Alternatively, incorrect clinical coding can be to the detriment of the informative nature of a complete diagnostic statement. Either way, the possibility of missing crucial information regarding comorbidities and complications experienced by the patient during the index admission cannot be ruled out.

Also, the event of having access to an EHR will enable a clearer picture of the patient's overall health for example, chronic medication, previous surgeries not occurring as part of the index admission or readmission and previous diagnoses and surgeries at other healthcare providers.

This study only considered admitting data and did not have clinical data such as blood pressure, heartrate, blood glucose levels, mobility scores, pain at discharge etc. to its availability. Access to data of this nature can provide the model with crucial information to improve prediction accuracy significantly.

5.5 RECOMMENDATIONS FOR FUTURE RESEARCH

In the clinical domain there exist numerous clinical indicators besides readmissions. For example, to name only a few, extended length of stay, mortality, septicaemia and heart failure. The prospects of modelling these indicators can transform the healthcare industry. Especially, if the predictions are real − time and thus patients are currently monitored and the moment if the probability of heart failure passes a certain threshold, notifications to clinical workers can provide an opportunity to prevent the heart failure from occurring. However, these studies have challenges with regards to data requirements, as well as, algorithms. The data of clinical measures such as heart rate needs to be accessible. Also, most of the clinical responses mentioned will suffer from class imbalance.

5.6 SUMMARY

The study successfully provides insight in the reason why clinical outcome monitoring and cost efficiency in healthcare is an ongoing phenomenon. The role that machine learning can play in this regard is highlighted and investigated throughout the study. Descriptions of four machine

learning models include a brief summary of both the underlying mathematical theory, as well as the advantages and disadvantages associated with each model. Details on the implementation of these models in SAS Enterprise Miner is discussed.

The study consists of 4 components, namely, data preparation, investigation of appropriate modelling algorithms, implementation of the algorithms on the constructed data and finally investigation of the model performance on training, validation and test datasets. Three modelling approaches was considered, which include, modelling on an unbalanced dataset with decision specification, modelling on a balanced dataset with prior probability specification and modelling on a balanced dataset with both prior probabilities, as well as, decision specification. Each approach presented a champion model, and the performance of each champion model on a test dataset is considered.

The data preparation step consisted of querying data from several base SAS datasets. Several variables included in the training dataset is not explicitly available in the data and required SAS Enterprise Guide's nifty and efficient data managing ability in order to construct the variables.

The final modelling datasets, as SAS dataset files, were provided to SAS Enterprise miner in a seamless fashion. The performance of the different models is uniform except for the SVM that did not perform well. The performance of neural networks was satisfactory. However, it is evident that the neural network performs better if it is preceded with a variable selection step. Despite the simple nature of decision trees, the decision tree algorithm illustrated that it deserves its place as an algorithm in machine learning applications. The logistic regression outperformed the other techniques. The study also described the utilisation of a readmission model in practice. The usage of the model includes benchmarking, as well as, risk stratification of patients in terms of risk for readmission. The golden threat that is visible throughout this study is the ability of machine learning algorithms to assist clinical workers and healthcare companies to become more efficient in its everyday operation.

APPENDIX A

SAS ENTERPRISE MINER WORKFLOW DIAGRAMS AND OUTPUT

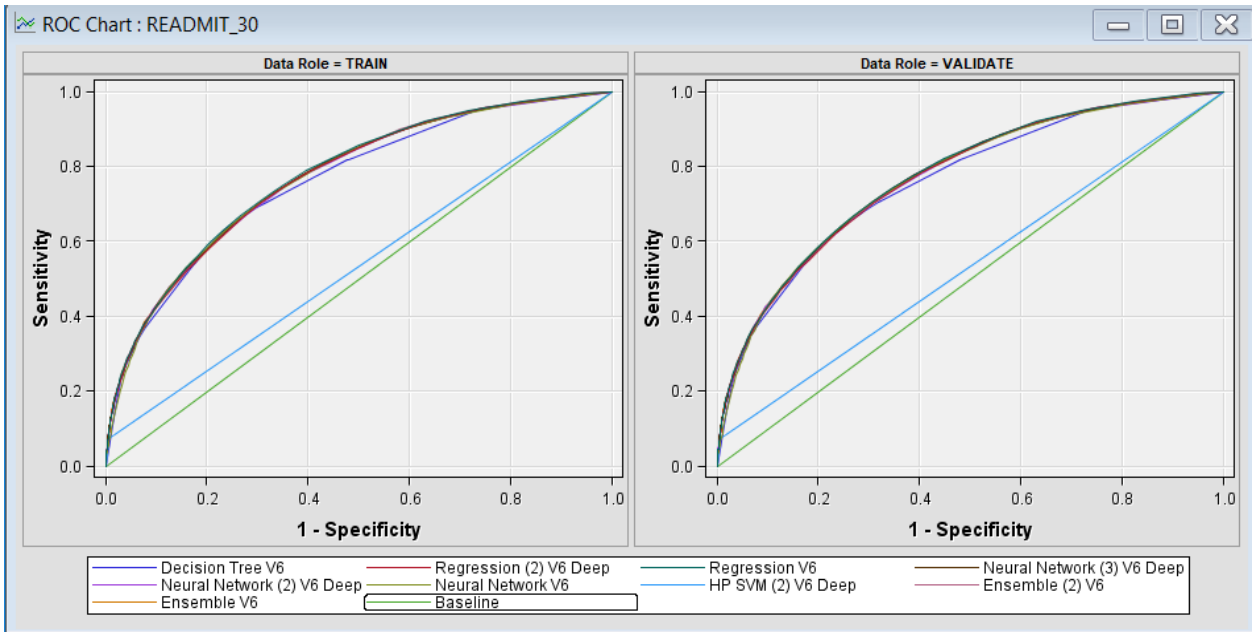

Figure A.1: ROC curves for models trained on unbalanced data with specification of cost matrix

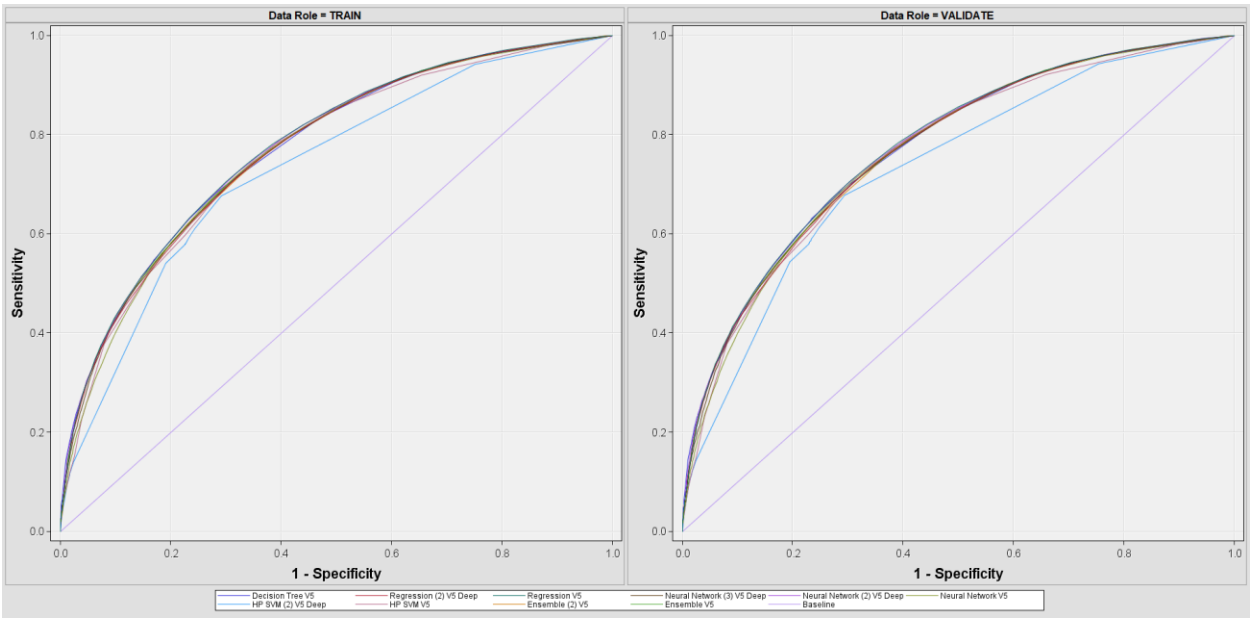

Figure A.2: ROC curves for models trained on balanced data with specification of prior probabilities

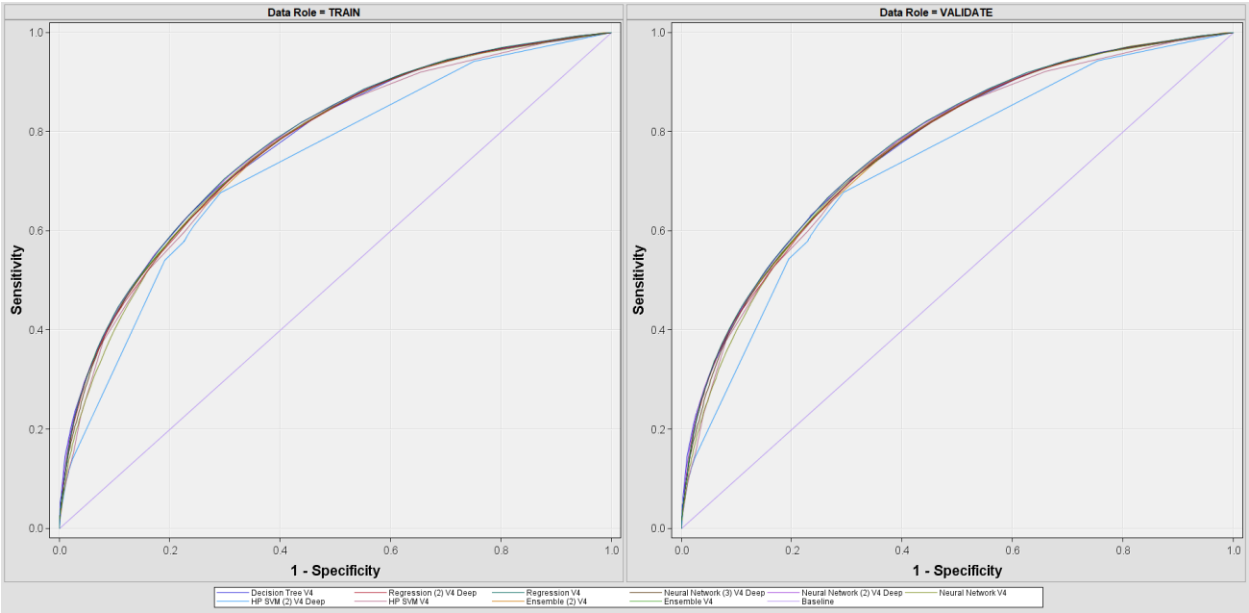

Figure A.3: ROC curves for models trained on balanced data with specification of cost matrix and prior probabilities

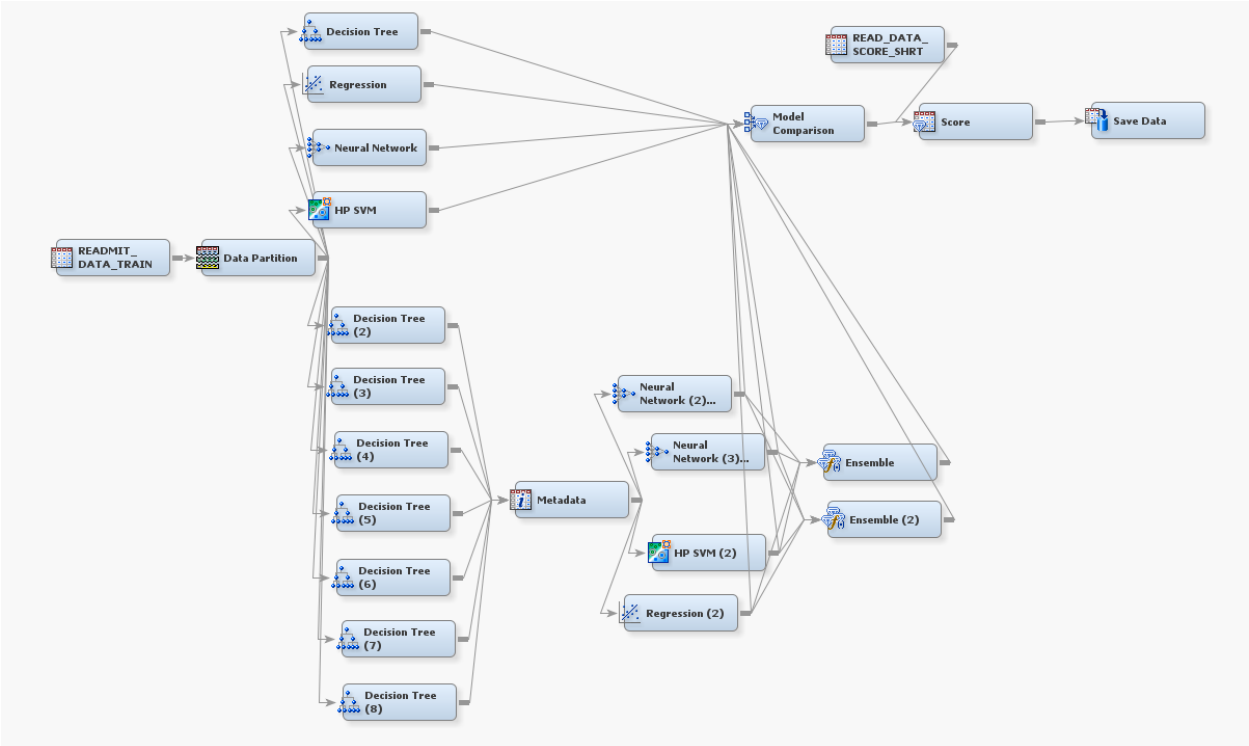

Figure A.4: Complete SAS Enterprise Miner workflow diagram

Bibliography

Agresti, A., 2002. *Categorical data analysis.* Second Edition. New Jersey: Wiley & Sons Inc.

Blanchard, R. & Wells, C. 2018. Deep learning using SAS® software. Corse Notes. *SAS Institute Inc.*

Billings, J., Blunt, I., Steventon, A., Georghiou, T., Lewis, G. & Bardsley, M. 2012. Development of a predictive model to identify inpatients at risk of re − admission within 30 days of discharge (PARR − 30). *BMJ open*, **2**, 1 – 9.

Brink. W., 2018. *Intro To Machine (And Deep) Learning, With A Focus On Probability And Uncertainty.* Deep learning Indaba X – Western Cape (UCT). Delivered April 2018.

Christie, P., Georges, J., Thompson, J. & Wells, C. 2015. Applied Analytics Using SAS® Enterprise Miner ™ Course Notes. *SAS Institute Inc*.

De Ville, B. & Neville, P. 2013. Decision Trees for Analytics Using SAS®Enterprise Miner. *SAS Institute Inc*.[Online]. Available at: https://support.sas.com/content/dam/SAS/support/en/books/decision-trees-for-analytics-using-sas-enterprise-miner/63319_excerpt.pdf. [27 June 2018].

Duggal, R., Shukla, S., Chandra, S., Shukla, B. & Khatri, S.K. 2016a. Impact of selected pre-processing techniques on prediction of risk of early readmission for diabetic patients in India. *International Journal of Diabetes in Developing Countries*, **36**(4), 469 – 476.

Duggal, R., Shukla, S., Chandra, S., Shukla, B. & Khatri, S.K. 2016b. Predictive risk modelling for early hospital readmission of patients with diabetes in India. *International Journal of Diabetes in Developing Countries*. Springer India, **36**(4), 519 – 528.

Futoma, J., Morris, J. & Lucas, J. 2015. A comparison of models for predicting early hospital readmissions. *Journal of Biomedical Informatics*. **56**, 229 – 238.

Gordon, L. 2013. Using classification and regression trees (CART) in SAS® Enterprise Miner™ for applications in public health. SAS Global Forum 2013. Data Mining and Text Analytics. *SAS Institute Inc.*

Hastie, T., Tibshirani, R. & Friedman, J. 2008. *The elements of statistical learning: Datamining inference and prediction*. Second Edition. Springer.

Longadge, R., Dongre, S.S. & Malik, L. 2013. Class Imbalance Problem in Data Mining: Review. *International Journal of Computer Science and Network*, **2**(1). Available at: https://arxiv.org/ftp/arxiv/papers/1305/1305.1707.pdf .[22 April 2018].

Marsland, S. 2009. *Machine learning: An algorithmic perspective.* Boca Raton: CRC. Chapman & Hall/CRC Machine Learning & Pattern Recognition Series.

Mazurowski, M. A., Habas, P.A., Zurada, J.M., LO, J.Y., Baker, J.A. & Tourassi, G.D. 2008. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, **21**, 427–436.

McIlvennan, C. K., Eapen, Z. J. & Allen, L. A. 2015. Hospital Readmissions Reduction Program. *Circulation*, ***131*** (20), 1796–1803.

Mediclinic International. 2018. *Operating Divisions*. [Online]. Available at: https://www.mediclinic.com/en/operating-divisions.html. [21 October 2018].

Mitchell, T. 1997. *Machine learning*. McGraw-Hill International Editions.

Neumann, A., Holstein, J., Le Gall, J.R. & Lepage, E. 2004. Measuring performance in health care: case − mix adjustment by boosted decision trees. *Artificial intelligence in healthcare.* **32**, 97 – 113.

Rice, J.A. 2007. *Mathematical Statistics and Data Analysis.* Third Edition. Brooks/Cole.

SAS Institute Inc., 2018a. SAS(R) 9.3 Language Reference: Concepts, Second Edition. *About SAS Date, Time, and Datetime Values.* [Online]. Available at: http://support.sas.com/documentation/cdl/en/lrcon/65287/HTML/default/viewer.htm#p1wj0wt2ebe2a0n1lv4lem9hdc0v.htm [4 August 2018].

SAS Institute Inc., 2018b. SAS(R) Enterprise Miner ™ 14.1Extension notes: Developers Guide. *Detecting Rare Cases.* [Online]. Available at: http://support.sas.com/documentation/cdl/en/emxndg/67980/HTML/default/viewer.htm#p1w6fewo0jhzxdn1rytuk1kt0pqj.htm [1 November 2018].

SAS Institute Inc., 2018c. SAS(R) Enterprise Miner ™ 14.1Extension notes: Developers Guide. *Prior probabilities.* [Online]. Available at: http://support.sas.com/documentation/cdl/en/emxndg/67980/HTML/default/viewer.htm#p1vqpbjwoo4bv7n1sw77e0z64xxs.htm [1 November 2018].

SAS Institute Inc., 2018d. SAS(R) Enterprise Miner ™ 14.1Extension notes: Developers Guide. *HP SVM Node.* [Online]. Available at: https://documentation.sas.com/?docsetId=emref&docsetTarget=n18ip3imet0wokn1f39nqoxy9138.htm&docsetVersion=15.1&locale=en [27 January 2019].

SAS Institute Inc., 2018e. SAS(R) Enterprise Miner ™ 14.1Extension notes: Developers Guide. *Neural Network Node: Usage.* [Online]. Available at:

https://documentation.sas.com/?docsetId=emref&docsetTarget=p1gqtpy080di3yn1d4i0xwlbe91h.htm&docsetVersion=15.1&locale=en [28 January 2019].

SAS Institute Inc., 2018f. SAS(R) Enterprise Miner ™ 14.1Extension notes: Developers Guide. *Neural Network Node: Reference.* [Online]. Available at: https://documentation.sas.com/?docsetId=emref&docsetTarget=p0zbqj1tu3h1uhn1x6regixbdg7v.htm&docsetVersion=15.1&locale=en [28 January 2019]

Shadmi, E., Flaks – Manov, N., Hoshen, M., Goldman, O., Bitterman, H. & Balicer, R.D. 2015. Predicting 30 − Day Readmissions With Preadmission Electronic Health Record Data. *Medical Care*, **53**(3), 283 – 289.

Shams, I., Ajorlou, S. & Yang, K. 2015. A predictive analytics approach to reducing 30 − day avoidable readmissions among patients with heart failure, acute myocardial infarction, pneumonia, or COPD. *Health Care Management Science*, **18**, 19–34.

Steel, S.J. 2017. *Assignment 6: Neural networks*. Lecture Notes. Statistical learning theory. University of Stellenbosch. Delivered October 2017.

Tong, L., Erdmann, C., Daldalian, M., Li, J. & Esposito, T. 2016. Comparison of predictive modeling approaches for 30-day all-cause non-elective readmission risk. *BMC medical research methodology*. 1-8.

Walsh, C. & Hripcsak, G. 2014. The effects of data sources, cohort selection, and outcome definition on a predictive model of risk of thirty − day hospital readmissions. *Journal of Biomedical Informatics*. **52,** 418–426.

Webb, A.R. (2002). *Statistical pattern recognition.* Second Edition. Wiley & Sons. Chichester.

WHO Collaborating Centre for Drug Statistics Methodology. 2018. [Online]. Available at: https://www.whocc.no/. [August 2018].

Yu, S., Farooq, F., van Esbroeck, A., Fung, G., Anand, V. & Krishnapuram, B. 2015. Predicting readmission risk with institution − specific prediction models. *Artificial Intelligence in Medicine*, **65**, 89 − 96.

Zheng, B., Zhang, J., Yoon, S.W., Lam, S.S., Khasawneh, M. & Poranki, S. 2015. Predictive modeling of hospital readmissions using metaheuristics and data mining. *Expert Systems with Applications*, **42**, 7110 – 7120.