Corporate failure prediction of JSE listed South African firms using machine learning

(A credit risk management approach)

Student name: Vukosi Era Maluleke

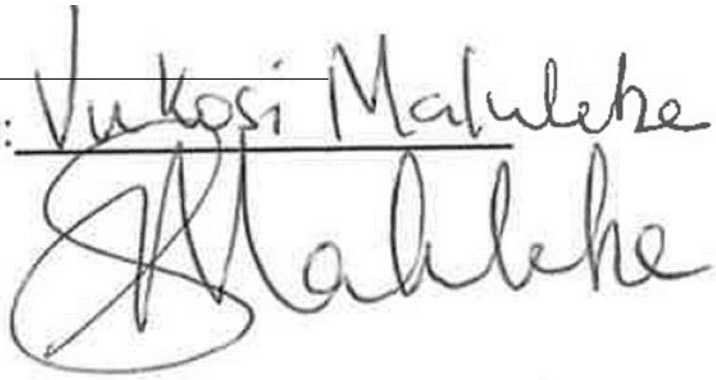
Supervisor: Dr. Joke Bührmann

A research dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, August 2020

Declaration

I declare that this research dissertation is my own unaided work. It is being submitted for the Degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

Signature: 

(Signature of Candidate)

26 day of August, 2020 in Roodepoort

Abstract

Corporate failure has increasingly become an area of interest to regulatory authorities, policy makers, lending institutions and private investors around the world. In South Africa, the deterioration of key macro-economic factors has raised the probability of credit defaults due to corporate failures. The aim of this study is to analyse, improve and expand prior work in the prediction of corporate failure in South Africa in order to develop a bankruptcy prediction model that can help predict corporate failure for listed companies in order to manage credit risk better. A sample of 366 companies listed on the JSE between 1997 and 2017 was used. This included 66 companies declared bankrupt during the period under investigation. Based on 72 financial variables, the results indicate that machine learning techniques outperformed statistical models by an additional 12% in prediction accuracy. The artificial neural network model yielded the highest accuracy score of 92%, followed by random forest models at 89% and support vector machine models at 87%. Results of this study may be of interest for financial institutions providing loans and for academics in machine learning prediction algorithms.

Acknowledgements

To God the Father who art in heaven, you deserve all the glory. To my supervisor, Dr. Joke Bührmann, this dissertation does not exist without you. Thanks for your patience, guidance and believing in me. Thanks to Mr Xolani Dastile for teaching me how to code in Python.

To my children Risima and Risana Maluleke, this work is dedicated to you. I hope it inspire both of you to a journey of many scientific discoveries. To my lovely wife Dr Xitshembiso Magomani-Mangolele, thanks for the support throughout this journey.

To my mother, N'wa-Risenga Mamaila Vumani and my uncle Mkhacane Samuel Vumani. Your prayers have sustained me throughout and I am forever grateful for your love and counsel. To my late father Hlengani Willie Maluleke and my late grandmother Mhlava N'wa-Khandli Vumani, you were the first to prophesy that one day I will become a "Dr" and it feels good to be one level below the title, I promise to fulfil your will. To all my siblings and the entire Maluleke-Mangolele family, with special mention to Florah Shihambani, Betty Masutha, Higgen, Nolly, Ndzalama, Tutu, Labane and Dr Themba Harry Mangolele, thanks for always being there for me.

Table of contents

List of figures

List of tables

List of acronyms

ANN Artificial Neural Network

BBR Bureau of Business Research

CART Classification and Regression Tree

CDF Cumulative Distribution Function

CFC Correlation Based Feature Selection

CFP Corporate Failure Prediction

CR Classification via Regression

ELU Exponential Linear Units

FCBF Fast Correlation-Based Feature Selection

JSE Johannesburg Stock Exchange

K-NN K-Nearest Neighbors

LBFGS Local Broyden Fletcher Goldfarb Shanno

LL Log-Likelihood

LOOCV Leave-One-Out Cross-Validation

MBF Markov Blanket Filter

MDA Multiple Discriminant Analysis

MDI Mean Decrease Impurity

MLP Multilayer perceptron

NBC Naive Bayes Classifier

PBIL Population-Based Incremental Learning

RBF Radial Basis Function

RELU Rectified Linear Unit

RF Random Forest

SBE Sequential Backward Elimination

SFFS Sequential Floating Forward Selection

SFS Sequential Forward Selection

SGD Stochastic Gradient Descent

SI Separability index

SME Small and Medium Enterprises

SVM Support Vector Machines

TANH Hyperbolic Tangent

1. Introduction

1.1 Research background

Corporate failure has increasingly become an area of interest to regulatory authorities, policy makers, lending institutions and private investors around the world (Sanaz *et al*, 2016). The prediction of corporate bankruptcy has been addressed as an increasingly important financial problem and extensively analysed in the accounting literature in South Africa (Candice, 2016). The deterioration of key macro-economic factors has raised the probability of increased credit defaults due to corporate failures (Douglas, 2016). This has triggered financial institutions like banks to become more proactive in the management of credit advancements to both individuals and companies.

Corporate bankruptcy or failure refers to an event where the company is unable to make profit or generate enough revenue to service its expenses or debts (Wagenmans, 2017). The success and failure of companies have a huge downstream effect to the economy including employment when the company does well and retrenchment when the company fails (Altman, 1968). The problem with lack of information on the likelihood of corporate failure makes it difficult to prevent bankruptcy from happening. The inability to predict corporate failure from key information remains a universal issue of concern for business and corporate organizations.

The pioneers to study the prediction of corporate failure were Beaver and Altman in 1966 and 1968 respectively (Altman, 1968) and (Beaver, 1966). Altman (1968) developed a z-score model which became a useful tool for predicting bankruptcy of a company. The z-score is calculated by a linear combination of several common financial ratios which are weighted by coefficients. The ratios cover aspects such as liquidity, profitability, operating efficiency, size and turnover of a firm (Altman, 2012).

Several studies in corporate failure and financial distress of companies were conducted afterwards using statistical methods. The 2008 global financial meltdown have recorded several cases of corporate failures worldwide (Douglas, 2016). This became a lesson to many financial institutions and policy makers to increase the effort of bankruptcy prediction in order to minimize credit losses resulting from this phenomenon.

Corporate failure causes unintended losses to various stakeholders with direct interest to the company. Commercial banks which extend credit to both firms and individuals stand to lose

the outstanding capital and interest. Firms carry out a lot of planning and apply different strategies in order to be as profitable as possible. Organisations sometimes take risks to achieve their goals that may lead to bankruptcy and big losses. The economy is adversely affected through job losses when a company files for bankruptcy (Sanaz *et al*, 2016).

Corporate failure prediction (CFP) is a popular problem investigated by researchers, credit institutions and banks (Fallahpour *et al*, 2017). Although extensive research has been conducted in this area, applications of combined variable selection methods and classification models are topics that have been given the most attention in recent years (Fallahpour *et al*, 2017). One of the most important issues in the bankruptcy prediction problem is to employ an effective variable selection algorithm which will lead to an acceptable level of performance accuracy in the implementation stage (Fallahpour *et al*, 2017).

Many studies on corporate failure prediction continue to be conducted around the globe but very little work is being done on this subject in South Africa (Douglas, 2016). Prior studies have been conducted using statistical techniques such as regression. Regression has been widely used since the mid nineteen sixties (Altman, 1968).

This study does not seek to reinvent the wheel but to expand and explore alternative ways to detect corporate failure using techniques such as artificial neural network, decision trees and support vector machine. The choice of machine learning is inspired by the fact that it has increasingly become a popular method of solving financial problems due to its ability to classify and learn from data (Goeletsis *et al*, 2012).

This research adds to the ongoing debate about the superiority of machine learning methods over statistical methods such as the one in Barboza *et al* (2017), Tsai *et al* (2014) and Yeh *et al* (2014). In addition to the five variables used by Altman (2012) in the z-score model, this dissertation introduces new variables including the enterprise value, operating margins, growth measures related to assets and sales, return on equity and others.

1.2 Research motivation

There has been growing interests in research undertakings in the topic of corporate failure after the 2008 financial crises where several cases were recorded of corporate failure in South Africa (Douglas, 2016). Another motivation for this study is to benefit accounting firms. If an accounting firm audits a potentially troubled firm and do not give appropriate warning signals it could faces costly lawsuits (Atiya, 2001).

Limited work has been published on South African corporate failure including the impact of credit defaults in the economy. According to (Douglas, 2016), the first credit default resulting from corporate failure in South Africa happened in 2013 by a company called First-Strut. This sparked interest in most commercial banks to pay attention to corporate failure and implement preventative measures. Recently, machine learning models have successfully been used for many classification problems and these models have often outperformed traditional classification methods (Bjorn *et al*, 2017).

1.3 Machine learning and corporate failure

Machine learning is a field of computer science that gives computer systems the ability to learn with data and improve from experience without being explicitly programmed. According to Bjorn (2017), since the early 1990s, machine learning techniques such as decision trees and neural networks have been widely used for both bankruptcy and credit default prediction Goeletsis *et al* (2012).

The machine learning field can be divided into three different categories: supervised learning, unsupervised learning and reinforcement learning (Raschka, 2016). Supervised learning is where the correct answers for a sample dataset are given and the model learns by example (Raschka, 2016). Unsupervised learning is when the model is not given any examples and needs to use clustering or similar techniques to analyse the given data. Reinforcement is where learning is obtained via the surrounding environment feedback and correct learned behavior is rewarded (Raschka, 2016).

Machine learning methods such as neural network, support vector machine and decision trees have proven to overcome several weaknesses of regression (Goeletsis *et al*, 2012). Their ability to learn from data coupled with their inherent technique of improving the learning error makes them popular to regression because machine learning methods are capable of identifying non-linear relationships in the data set.

There are two major types of supervised machine learning problems, called classification and regression. In classification, the main aim is to predict a class label, which is a choice from a predefined list of options. Classification is sometimes distinguished into binary and multiclass classification, where binary classification is the special case of distinguishing between two classes, while multiclass classification is distinguishing between more than two classes (Simon, 2009).

Simon (2009) explains that in regression modelling, the aim is to predict a continuous number or a floating-point number. Predicting a person's monthly income from their level of education, college, their age, and address is an example of a regression task.

An easy way to distinguish between classification and regression tasks is to ask whether there is some element of continuity in the output. If there exists a property of continuity between possible outcomes, then the problem is a regression problem (Simon, 2009). In this study, classification supervised machine learning is used to predict whether a company will go bankrupt within the next 12 months from the financial results analysed.

Three machine learning techniques relevant to this study are discussed in more details. These are artificial neural network, decision trees and support vector machines. The machine learning techniques to be used in this study falls under supervised learning. Supervised learning is where dataset is given the correct answers, i.e. bankrupt or not for a sample dataset and learns by example.

In this study classification supervised machine learning models are used to predict whether a company will go bankrupt within the next year from the financial statements analysed. This is done through the applications of combined variable selection methods and classification models including both regression models and machine learning techniques.

1.4 Research question

The research question this dissertation seeks to answer is how effective can machine learning be applied in the credit risk management space through developing models to predict bankruptcy using neural network, support vector machines and decisions tree. Commercial banks are required to study the quality of potential borrower's credit worthiness in order to make their decision on corporate loans, as a result the outcome of this study will be useful for commercial banks to understand credit management better.

1.5 Research objectives

The research objectives are as follows:

1. Investigate and compare the prediction accuracy of the following models:
    a. Logistic regression model
    b. Artificial Neural Networks
    c. Decision trees classifications
    d. Support Vector Machine

2. To determine the machine learning model (from those stated in item 1) that can accurately predict bankruptcy.

The objective of the study is therefore to investigate how the machine learning techniques: ANN, Decision trees, and SVM can be exploited in the field of finance, specifically in credit risk management.

1.6 Research purpose

The purpose of this research is illustrate how machine learning techniques can be used to predict corporate failures compared to the statistical regression models and to determine the best technique to use out of the machine learning methods. This was done in order to analyse, improve and expand prior work in the prediction of corporate failure in South Africa and to develop a bankruptcy prediction model that can help predict corporate failure in order to manage credit risk better. This is achieved by developing machine learning based classification algorithms including artificial neural network, decision trees, and support vector machine models which can classify financially well-off companies from companies about to go bankrupt.

1.7 Significance of the Study

Corporate failure has become a universal concern to all stakeholders (i.e. Investors, customers, senior managers) due to its social downstream impact on the economy. Successful prediction of this event can lead to robust preventative measures which can boost company revenues and elevate economic growth (Atiya, 2001). This investigation will enable policy makers and regulators, especially at commercial banks to make informed decisions when assessing the credit worthiness of a corporate for investment purposes.

Predicting corporate failure in a manageable time-window can help troubled firms to take appropriate measures earlier and therefore helps them avoid liquidation and prevent job losses. This can also help lenders to be more proactive and take appropriate actions to prevent financial losses resulting from bankruptcy from the borrowing companies.

This research will therefore be useful and relevant to the management of credit risk in financial institutions. Beyond this, the research also contributes to the existing debate on the superiority of machine learning methods over statistical method in bankruptcy prediction and can be used to identify which financial factors are relevant in making the predictions.

1.8 Report layout

The dissertation is organized into six chapters. Chapter 2 is the literature review, this chapter outlines prior research on the subject of corporate failure and the accompanied findings. It also investigates corporate failure using both statistical means such as binary logistic regression and various machine learning methodologies including artificial neural network, support vector machine and decision trees. The chapter also covers a detailed background of each prediction methodology.

Chapter 3 explains different machine learning techniques used in bankruptcy prediction. The chapter gives the different types of learning applicable in machine learning which includes supervised learning, unsupervised learning, and reinforcement learning. This chapter also gives the detail information about the three different machine learning techniques used in this study which includes artificial neural network, decision trees and support vector machines.

The fourth chapter explains the methodology and data. This chapter examines the data and methodology which is employed in this study. The chapter also explains the data and defines the data split in terms of training sample, which is the development data set, the test sample which is the validation data set, and the actual data set which represent new data which the

model was not built or validated upon. All the models are built using the training sample while their performance is tested on both the validation and actual data set.

The fifth chapter discusses the results and analysis. This chapter explains and analyse the results from all the models. The first part explains and covers the outcome of each model and its predictive power which measures the goodness of fit in the form of an accuracy ratio, the higher the goodness of fit the better the model in separating good companies from bad companies. Next, the chapter analyses the predictive power of the models based on varying the parameters of the models.

Chapter 6 draws conclusions based on the results and analysis. This chapter contains a short summary of the entire research work and the main findings. The chapter also addresses how the research question is answered in terms of the results and analysis. The chapter proceeds to explain which method between statistical analysis and machine learning predicts bankruptcy the best for the case study. Future areas of research are also identified.

The appendices contain the python codes used to develop the various machine learning models including ANN, SVM and random forest used in this study. This section also contains all the financial variables used to build the prediction models and their definitions. The data was source from IRESS expert while the codes were developed in python programming language.

2. Literature review

The literature study is divided into three parts. The first part is covered in section (2.1) to (2.4), this part gives background about corporate failure, its different definitions, pioneer studies including both the historical and modern studies in the topic. The second part is the application of statistical methods in corporate failure prediction which is covered from section (2.5). Lastly the last part is application of machine learning in corporate failure which is covered in section (3).

2.1 Definition of corporate bankruptcy

Altman (1968) and Beaver (1966) were the first to define bankruptcy, Altman (1968) has defined bankruptcy to be an instance where the rate of return on the capital invested in the company is less than the rates on similar investments. On the other hand, Beaver (1966) defined bankruptcy as an instance where a company defaulted on a loan obligation or missed preferred dividend payments. Elloumi and Gueyie (2001) define bankruptcy as a condition whereby a firm's business deteriorates to the point where it cannot meet its financial obligations. The following section looks at the existing formal definitions of bankruptcy and the one chosen to be used in this study

2.1.1 Definition of bankruptcy in this study

In South Africa the definition of bankruptcy or financial distress has been described as those companies that were forced to delist from the Johannesburg Stock Exchange (JSE) due to poor financial performance leading to liquidation (Radloff and van der Walt, 1999). A more universal definition by Bruwer & Harmann (2009) describes bankruptcy as a company that will not survive in its existing structure, and therefore encompasses a delisting or a major structural change.

As noted by Karels and Prakash (1987), there is no single definition of bankruptcy across the world. There is a wide set of definitions of failure used for prediction studies (Jodi *et al*, 2007). Many studies define corporate failure as the actual filing for bankruptcy or liquidation in the court of law. However, others define bankruptcy as suffering financial stress or an inability to pay financial obligations. Some studies do not provide the definition of failure used for the research (Jodi *et al*, 2007).

The study adopted the definition of bankruptcy from Frances (2017). The author explained that for a company to fall into the category financial distress, one of the following conditions needs to have occurred:

1. A company may undergo financial restructuring that it would not have taken if it had sufficient Cashflow (Rao, 2017).

2. Asset Restructuring - A company has sold off its assets to repay debt, this usually happens when the value of the equity is less than the debt payment as the point of The debt obligation (Gaughan, 2015).

3. A company has been suspended from trading on the stock exchange (in this case the JSE). In this study all suspended companies selected would have been selected because they were suspended because they did not submit financial statements as required as part of its listing requirements

4. A company has successfully filed for liquidation through a court of law.

5. A company has voluntarily delisted in order to create shareholders value, this usually happens when management decides to protect the value of shareholders from an undervaluation of the company share by the market (Pour, 2013).

6. The share price of the listed company has dropped by more than half.

The next section explains some of the first earlier studies in literature including the history of corporate failure prediction.

2.2 Pioneer studies in corporate failure prediction

Altman (1968) and Beaver (1966) investigated corporate failure using multiple discriminant analysis of financial ratios. A cut–off level score was determined as a separating decision between bankrupt and non-bankrupt companies. Since then the model has been well received as a business failure prediction model.

Altman (1968) investigated the balance sheet and income statement of various companies using solvency, liquidity and profitability ratios extracted from financial results as inputs to the model. Altman (1968) developed a logistic regression model commonly known as the z-score Altman model. Logistic regression or the logit model is an extension of simple linear regression which analyses the relationship between multiple independent variables and a categorical

dependent variable (Park, 2013). The model estimates the probability of occurrence of an event by fitting data to a logistic curve (Park, 2013).

2.3 History of bankruptcy prediction

The early studies of bankruptcy predictions were univariate studies (Jodi *et al*, 2007). These studies focused on single ratios by comparing the individual ratios of failed companies with those of successful companies. The same authority argued that this univariate models were a ground breaker and paved way to multivariate prediction models seen in recent times. They continued to argue that studies around bankruptcy prediction happened as early as 1930's when the Bureau of Business Research (BBR) published a bulletin with results of financial ratios of financial distressed industrial firms.

Fitzpatrick (1932) conducted a study using 13 financial ratios and 19 non-financial ratios of distressed companies. The study found that non-distressed companies displayed good financial ratios than distressed companies. Fitzpatrick (1932) highlighted that the two most predictive ratios were the "Net worth to Debt ratio" and "Net worth to Profit ratio" while the less predictive ones were the "Current ratio" and "Quick ratio".

Following the publication by BBR in 1930, Winakor and Smith (1935) used a sample of 183 financially distressed firms from different sectors of the economy. The results of their study was that "Working capital to total assets ratio'' was a better predictor of financial distress that both "Cash to Total Assets ratio" and "Current ratio". Another interesting finding of their study was that a higher value for "Current Assets to Total Assets" decreases the company's likelihood of bankruptcy, this makes sense as one will practically expect the assets to decrease and debt levels to increase as the company faces financial difficulties.

Merwin (1942) conducted a study of bankruptcy prediction focusing on small manufacturing firms. His study found that when comparing the financial ratios of financially distressed and non-distressed companies, the financially distressed companies display signs of weakness as early as four to five years before the company fail. Merwin (1942) highlighted that the most predictive factors were the "Net Working Capital to Total Assets ratio", "Current ratio'' and "Net Worth to Total Debt ratio".

Chudson (1945) conducted a bankruptcy prediction study across different industries. The aim of his study was to determine if there was a "normal" pattern to a financial structure of financially distressed firm. Chudson (1945) found that there is evidence of clustering amongst

other financial ratios such as “Size” and “Profitability” in certain industries. However, his overall finding was that there was no “normal” pattern to the state of financial distress while comparing ratios of financial distressed firms and non-financial distressed firms.

There has been substantial literature contribution to this subject of bankruptcy prediction over the past 40 years (Frances, 2017). Jodi *et al* (2007) wrote a paper which traces the literature on bankruptcy prediction from the 1930’s. In their study, the authors explained the evolution of bankruptcy prediction studies over time including the different methodologies, variety of prediction factors and the specific uses of the models. The authors used a total of 165 bankruptcy prediction studies published from 1965 to 2007 (Jodi *et al*, 2007).

The analysis on the prediction accuracy from Jodi *et al* (2007) suggested that neural network together with multivariate discriminant analysis were the most efficient methodologies to predict bankruptcy. The study also found that great number of factors does not always translates to high model accuracy, the study has evidence that some models with only two factors were as good as models with more than twenty factors.

Edmister (1972) developed a bankruptcy model for small businesses. The purpose of his study was to test the efficiency of using financial ratios in predicting small business failure. His study was different to the one of Beaver (1966) and Altman (1968) because both beaver and Altman used univariate and discriminant analysis respectively on medium and large firms. Edmister (1972) argued that the Beaver (1966) and Altman (1968) ignored small businesses because of the difficulty on obtaining data.

Deakin (1972) conducted a study where he compared both the Beaver univariate model and the Altman multiple discriminant model using the same choice of variables. However, Deakin (1972) used a different definition of bankruptcy compared to the one used by Beaver (1966). Deakin (1972) definition of bankruptcy was only limited to both insolvency and liquidation whereas Beaver (1966) definition of bankruptcy covered insolvency and liquidation, as well as actual default on loan obligations. Deakin (1972) empirical finding showed that the discriminant analysis was a better predictor of corporate bankruptcy than the univariate analysis.

Sinkey (1975) developed a model aimed at predicting bank failure. The study presented a multivariate statistical analysis model on the balance sheet and income statement of financially distressed banks. The model makes use of the multiple discriminant analysis (MDA) to create classification rules based on the group mean differences between a financial distressed bank

and a non-financial distressed bank. The empirical finding showed that banking factors such as asset composition and capital adequacy amongst other factors are good discriminators between financial distressed and non-financial distressed banks.

Jackendof (1962) conducted a research to compare the financial ratios of profitable and non-profitable firms. The study found that the two ratios, namely “Current ratio” and “Net Working Capital to Total Assets ratio” were higher for profitable firms than for non-profitable firms. Another finding was that profitable firms had lower debt to worth ratios than non-profitable (Jodi *et al*, 2007).

These early studies laid the groundwork for the studies that followed especially the univariate model by Beaver (1966) and multiple discriminant analysis by Altman (1968). Beaver (1966) continued with the study and tested the individual ratios predictive power in classifying bankruptcy. The study found that the “Net Income to Total Debt ratio” had the highest predictive power, followed by “Net Income to Sales ratio”, “Cash Flow to Total Debt ratio” and “Cash Flow to Total Assets ratio” respectively (Jodi *et al*, 2007).

The first multivariate study was published by Altman in 1968 (Jodi *et al*, 2007). Altman (1968) used multivariate discriminant analysis to develop a five-factor model to predict bankruptcy of manufacturing firms. The "z-score", as it was called, predicted bankruptcy if the firm's score fell within a certain range (Jodi *et al*, 2007). The model's predictive ability when tested on a hold-out sample was 79% (Jodi *et al*, 2007).

After the Altman’s model, there has been an increased number of bankruptcy prediction studies. According to Jodi *et al* (2007), after Beaver (1966) and Altman (1968), the numbers of published studies in bankruptcy prediction climbed from two studies in the 1960’s to 28 studies in the 1970's; 53 studies in the 1980's and 70 studies in the 1990's.

2.4 Modern studies in the subject of corporate bankruptcy

Today, the intensity of industry competition has led many companies going bankrupt and pulling out of race (Kasgari *et al*, 2013). Altman (1968) and Beaver (1966) investigated corporate failure using linear regression on financial ratios, though they were not the first to study the topic of bankruptcy, they were the first to study bankruptcy using statistical and mathematical models.

Kim *et al* (1993) conducted a study to classify bonds in financial investments based on the possibility that the borrowing company may default on its debt obligation. Kim *et al* (1993)

compared the neural network with statistical methods covering discriminant analysis, logistic analysis, and a rule-based system for bond rating. The findings were that neural networks were better classification techniques compared to regression techniques in terms of the accuracy rate.

Al-Osaimy (1998) conducted a study to predict the performance of Islamic banks. The study used a seven factor neural network model for predicting Islamic banks performance. The findings of the study were that neural network methods showed significant performance for providing early warning signals for various types of performance including bankruptcy, insolvency and failure (Frances, 2017). The study also correctly classified 89% of the banks on the two years prior data set which was unseen by the network (Al-Osaimy, 1998).

Douglas (2016) conducted a study to predict corporate failure on corporates using multiple discriminant analysis. The model resulted in 82.95% prediction rate in terms of the overall accuracy rate for correctly classifying financially distressed and non-financial distressed firms. Bjorn (2017) used both machine learning methods and statistical methods to predict corporate failure using Polish companies. The outcome indicated that machine learning outperformed regression models.

Francois (2016) conducted a study to assess whether a model based on both qualitative and quantitative financial factors can predict corporate failure better than a quantitative financial model. The results showed evidence of leverage in combining qualitative variables with quantitative variables in predicting corporate failure. Candice (2016) employed a decision Classification and Regression Trees (CART) algorithm to develop a financially distressed prediction models for South African firms which resulted in more accurate classification.

Konirk (2004) conducted a study to assess corporate failure using unsupervised machine learning algorithms such as population-based incremental learning (PBIL) and the k-nearest neighbor's algorithm (k-NN). PBIL is a method of combining genetic algorithms and competitive learning for function optimisation (Baluja, 1994). The (k-NN) algorithm is a non-parametric clustering method used for classification and regression.

After running PBIL numerous times, the algorithm successfully separated companies into failed companies and good companies. Konirk (2004) used the Thornton's Separability index (SI) to evaluate the degree of separation achieved by each feature subset. The index showed good separation results. The k-NN classifier algorithm also successfully showed good separation but was outperformed by the PBIL classification algorithm.

Danenas and Garsva (2015) built a bankruptcy prediction model using artificial neural network combined with fuzzy logic. Their study opened further opportunities for predicting bankruptcy on companies that are showing signs, artificial neural network yielded an accuracy of 88% while the fuzzy logic demonstrated superiority with 94% accuracy.

Sanchez *et al* (2016) conducted a study to predict bankruptcy by comparing the effectiveness of the hybrid neural network method called the associative memory translation with traditional neural networks, a support vector machine and a logistic regression model in terms of their prediction capabilities. The experimental results over nine real-life financial databases suggested that the associative memory is the appropriate approach to prediction of financial distress (Sanchez *et al*, 2016).

Wang and Li (2018) conducted a study to compare statistical methods and machine learning methods to predict corporate bankruptcy using Chinese listed companies. The empirical results showed that machine learning methods are superior to statistical methods. The Accuracy of the study showed a 95.9% prediction rate which was higher than existing studies.

Barboza *et al* (2017) conducted a study where they tested machine learning models such as support vector machine and random forest on North Americans firms. Machine learning models showed an over performance of 10% more than traditional models, with random forest at 87% accuracy and logistic regression at 67% accuracy.

Murat and Kermal (2014) used qualitative variables on machine learning to predict bankruptcy. Qualitative bankruptcy prediction methods represent experts problem-solving knowledge to predict qualitative bankruptcy of a particular company. Murat and Kermal (2014) used four different machine learning techniques for qualitative bankruptcy prediction, namely; Naive Bayes Classifier (NBC), Multilayer Perceptron (MLP), J48 and Classification via Regression (CR). The outcome of the study showed that correctly classified cases were found to be very accurate between 94.86% and 96.57%, with NBC performing the best.

Other several studies have been developed using machine learning. These methods are technology reliant and predict bankruptcy by learning from the input data without any prior assumptions regarding the distribution of the input variables or the error term (Goeletsis *et al*, 2012). The most common techniques include from artificial neural network, support vector machine, and decision trees.

According to Candice (2016), when bankruptcy prediction first became popular, the most common methods used to build the models were Multiple Discriminant Analysis (MDA), Logit analysis as well as Probit analysis. However, financial distress prediction models and advanced machine learning techniques were introduced to the subject. Machine learning techniques became the preferred method due to their ability to self-learn from the data without any prior distributional assumptions. (Goeletsis *et al*, 2012).

The literature study has been divided into two sections, i.e. predicting corporate failure using statistical techniques and predicting corporate failure using machine learning. The first section applies binary logistic regression to classify bankrupt and non-bankrupt companies. The second section summarizes different classification methods using machine learning based models such as ANN, Decision trees, and SVM.

The next section explains the logistic regression model, the logistic model is presented as the benchmark model in this study. After the logistic regression section, machine learning methods such as neural network, support vector machine and decision trees are investigated and compared to the benchmark model in order to establish which method of prediction is the most effective by comparing the accuracy of the two methods.

2.5 Application of statistical methods in bankruptcy prediction

2.5.1 The Altman regression model

Altman was the pioneer to develop a multivariate logistic regression model to predict bankruptcy, called the Altman z-score model (Altman, 1968). The model uses stepwise logistic regression to examine a combination of financial ratios from USA companies. The Altman regression model is popular in the management of credit risk (granting and rejection of debt funding) Altman (2012). The Altman model outputs a credit score. This output z-score represents the strength of credit which estimates the company's probability of defaulting which will result in corporate failure.

Altman (1968) analysed sixty-six USA companies where the split between bankrupt and non-bankrupt was 50/50 (i.e. 33 bankrupt firms and 33 non-bankrupt firms). All the companies were from the manufacturing industry (France, 2017). The study then used stepwise regression to examine a combination of financial ratios in order to determine the best possible combination in terms of predictive power. A multiple discriminant function was calculated, this is called the z-score model and it is composed of five financial performance ratios.

Based on other studies, Altman (1968) came up with a subset of twenty-two financial variables. He then went on to perform a ranking analysis of the twenty-two financial ratios. The final variables were narrowed down to five factors. Other variables such as "Cashflow to Debt ratio" and others were ignored in his model because of gaps in the historical data.

$$z_{score} = 1.2x_1+1.4x_2+3.3x_3+0.6x_4+0.999x_5, \quad (1)$$

The statistical model estimates the weights to use in the of the z-score model where

x_1 is the working capital / total asset (Measures liquidity);

x_2 is the retained earnings / total assets (Measures profitability);

x_3 is the earnings before interest and tax / total assets (Measures operating efficiency);

x_4 is the market value of equity / total liabilities (Adds market size) and;

x_5 is the sales / total assets (Measure for turnover)

When analyzing the final z-score of a company, the lower the score the higher the probability of business failure. Altman (1968) defined the following rules for interpreting a firm z-score:

i) below 1.8 indicates a firm is heading for bankruptcy;

ii) above 3.0 indicates a firm is unlikely to enter bankruptcy; and

iii) between 1.8 and 3.0 is an indecisive area

The Altman model to predict corporate failure is limited due to assumptions such as linearity, normality and independence among predictor variables (Goeletsis *et al*, 2012). The weakness in regression techniques causes limitations in obtaining effective and accurate results. Over recent years, several machine learning methods have been effectively applied to build accurate predictive models for detecting business failure with remarkable results, such as neural networks and ensemble methods and there is an on-going debate about the superiority of machine learning methods over statistical methods (Deakin, 1976).

Although the Altman z-score model is widely used, it was developed specifically for USA financial indicators. In this study, a new logistic regression model was developed to use financial indicators provided by JSE listed companies. In addition to the five variables used by Altman (1968), this model introduced new variables including the enterprise value, operating margins, growth measures related to assets and sales, return on equity and others. The next

sections explain various logistic regression techniques investigated including the Univariate analysis model, Multivariate discriminant analysis model, the Probit model and Logit model.

2.5.2 Multivariate regression model

The multivariate regression model is a statistical process for estimating the relationships between a dependent variable and one or more independent variables (Frances, 2017). The first multivariate study was published by Altman (1968). For the purpose of bankruptcy prediction, there are only two outcomes for the dependent variable, "Bankrupt" or "Non bankrupt". The dependent variable is a binary variable that has two outcomes, either a one for bankruptcy or a zero for non-bankruptcy. The independent variables are accounting variables or financial ratios from the balance sheet, Cashflow statement and income statement.

Frances (2017) have classified multivariate regression models into at least four different types, namely;

1. The Univariate analysis model,
2. The Multivariate discriminant analysis model,
3. The Probit model and
4. The Logit model.

The most popular regression methods have been the discriminate analysis followed by the Logit analysis (Altman and Saunders, 1998).

2.5.3 Univariate analysis and multivariate analysis models

A regression model expresses the relationship between the dependent variable and the independent variable. Usually the regression model expresses the relationship between the response variable as a function of the independent financial ratios with an error term.

The dependent variable of a univariate regression model is expressed as a function of single variable and an error term:

$$Y(x) = \beta_0 + \beta_1 x_1 + \epsilon \,, \qquad (2)$$

Where Y is the response variable,

x_1 is the independent financial variable,

β_0 is a constant, and

β_1 is a regression co-efficient and ϵ is the error term.

The discriminant analysis was pioneered by Fisher (1936). For multiple discriminant analysis, the regression is modelled in terms of more than one variable. The equation of a multiple regression model takes the following form:

$$Y(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots\ldots\ldots \beta_n x_n + \epsilon, \quad (3)$$

Where Y is the response variable,

$x = x_1, x_2, x_3, \ldots\ldots x_n$ is the independent financial variable,

β_0 is a constant, $\beta_1, \beta_2, \beta_3, \ldots\ldots \beta_n$ are a regression co-efficient and

ϵ is the error term.

According to Frances (2017), the major underlying assumptions of discriminate analysis are as follows:

1. The observations are a random sample with each predictor variable normally distributed,

2. Each of the allocations for the dependent categories in the initial training set is correctly classified,

3. There must be at least two groups or categories, with each case belonging to only one group so that the groups are mutually exclusive.

According to Frances (2017), the discriminate analysis is a form of supervised learning. A definition of this has already been provided in above. The main aim of discriminant analysis is to capture the underlying information of the independent variables (i.e. $x_1, x_2, x_3, \ldots\ldots x_n$) in a mathematical way such that the new composite variable called the Bankruptcy function Y(x) is determined.

The next section explains the advantages and disadvantages of both the Univariate and Multivariate regression models.

2.5.3.1 Advantages of univariate analysis and multivariate analysis regression models

According to Frances (2017), the advantages of the two types of methods includes the following

1. Easy to implement in both excel or any programming language and
2. Easy to interpret the final results.

2.5.3.2 Disadvantages of univariate analysis and multivariate analysis regression models

1. The multivariate regression needs strong assumptions that the distributions of predictors within each are normally distributed (Frances, 2017),
2. Unordered categorical predictors cannot be used as the input variables (Frances, 2017),
3. Multivariate regression analysis is very sensitive to outliers than logistic regression (Frances, 2017).

2.5.4 Probit regression models

Frances (2017) explains the Probit model as the one which predicts a binary dependent variable on a given set of input variables. The Probit model makes use of conditional probabilities p(x). The probability of the Probit model can be captured by the cumulative distribution function (CDF) of the standard normal distribution (Frances, 2017). The main difference between the Probit regression model and the logit regression model is that the conditional probability p(x) of the logit model can be captured by the CDF of the logistic distribution and the conditional (Frances, 2017).

The conditional probability of the Probit model is given by the following equation:

Let $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots\dots\dots \beta_n x_n$, (4)

Then $Pr\ (Y=1\ /X=x) = p(x) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-0.5t^2} d_t$, (5)

The advantages of Probit regression models (Frances, 2017):

1. The Normal distribution of independent variables is not a requirement
2. The unbounded of independent variables is not a requirement.

The disadvantages of Probit regression models (Frances, 2017):

1. There is no assumption of a linear relationship between the dependent variables and the independent variables but it assumes a linear relationship between the logit transformation of the dependent variables and independent variables

2. The dependent variable has to be binary.

The next section explains the logistic regression in detail including its assumptions and set-up. The logistic regression is the benchmark model in this study, hence its methodology is covered in depth. The logistic regression benchmark model is implemented using the methodology explained below.

2.5.5 Logistic regression

Logistic regression is a type of regression used to fit data on a regression equation to make future predictions. Various logistic regression techniques were investigated including the Univariate analysis model, Multivariate discriminant analysis model, the Probit model and Logit model.

Logistic regression was first proposed in the late 1960's and early 1970's and has since become the most common method of regression since 1980s (Peng & Lee, 2002). The focus is on binary logistic regression since the problem is a binary classification problem. Binary logistic regression is generally used when the dependent variable is dichotomous and the independent variables are either continuous or categorical (Park, 2013).

2.5.5.1 Assumptions of logistic regression

Logistic regression is a type of regression which uses the Logit transformation to linearize the non-linear relationship between input variables x and the probability of y (Park, 2013). In logistic regression, a mathematical model of a set of explanatory variables is used to predict a *logit* transformation of the dependent variable. However, there are several problems which must be overcome before regressing with a dependent variable which is dichotomous or binary in nature. The following assumptions must hold for logistic regression (Park, 2013):

1. The dependent variable is quantitative, continuous (i.e. can take on any numerical value), and unbounded. If the dependent variable does not meet these requirements, then predicted output on the dependent variable may lie outside possible limits
2. The relationship between the dependent variable and independent variables is linear and additive
3. The error term is normally distributed
4. The variance of the error term is constant across all values of the independent variables.

2.5.5.2 Logit transformation

Since logistic regression calculates the probability of an event occurring compared to the probability of an event not occurring, the impact of independent variables is usually explained in terms of odds (Park, 2013).

2.5.5.3 Odds of event

Odds of an event are the ratio of the probability that an event will occur to the probability that it will not occur. If the probability of an event occurring is p, the probability of the event not occurring is (1-p) (Park, 2013). Then the odds of the event are:

$$\text{Odds of event} = \frac{P}{(1-p)}, \quad (6)$$

The logit model uses the natural logarithm to transform a non-linear relationship between input variables $x_i$and the probability of event Y happening to a linear relationship:

$$Y = \sum_{i=1}^{m} \alpha + b_i x_i, \quad (7)$$

where m is the number of independent variables,$\alpha$is the constant term of the regression model (i.e. the mean expected outcome), x_i is the value of the independent variables and b_i is the coefficient of the independent variable. However the transformation is not a good model because extreme values of $x_i$gives values of$\alpha+b_i x_i$ which does not fall between zero and one. The solution to this problem is to transform the odds using the natural logarithm (Park, 2013):

$$\text{Logit (Y)} = \ln(\text{odds}) = \ln\left(\frac{P}{(1-p)}\right) = \alpha + bx_i, \quad (8)$$

where p the probability of the response variable Y. In this case the probability of corporate failure and x_i is the explanatory variable of the feature attribute (Park, 2013). The parameters of the logistic regression are α and b. Eq. (8) represents a logistic model (Park, 2013). Applying the antilog operator on Eq. (8) on both sides, one can deduce an equation for the prediction of the conditional probability of the occurrence of interested outcome as shown by the following equation:

$$p = P(Y=Y'/X=x) = \frac{e^{\alpha + bx}}{1+e^{\alpha + bx}} = \frac{1}{1+e^{-(\alpha + bx)}}, \quad (9)$$

where Y' is the outcome of interest for Y(Park, 2013). Applying the logic of the simple logistic regression to multiple predictors, derives to the following equation:

$$\text{Logit (Y)} = \ln(\text{odds}) = \ln\left(\frac{P}{(1-p)}\right) = \alpha + b_1x_1 + b_1x_2 \dots \dots \dots b_mx_m \,, \quad (10)$$

where m is the number of independent variables,$\alpha$is the constant or mean (i.e. the expected outcome) (Park, 2013). Therefore, deduce from Eq. (10) the following equation:

$$p = P\,(Y = Y'\,/X = x) =$$

$$\frac{e^{\alpha + b_1x_1 + b_1x_2 \dots\dots\dots b_mx_m}}{1+e^{\alpha + b_1x_1 + b_1x_2 \dots\dots\dots b_mx_m}} = \frac{1}{1+e^{-(\alpha + b_1x_1 + b_1x_2 \dots\dots\dots b_mx_m)}} \,, \quad (11)$$

2.5.5.4 Odds ratio and log likelihood function

The odds ratio (OR) is a comparative measure of two odds relative to different events (Park, 2013). For two events A and B, the corresponding odds of event A occurring relative to event B occurring is given as below:

$$\text{Odds Ratio} = \text{OR} = \frac{\text{Odds}\{A\}}{Odds\{B\}} = \frac{P_{A/(1-P_A)}}{P_{B/(1-P_B)}} \,, \quad (12)$$

OR is a measure of the relationship between an outcome and an event (Park, 2013). For the purpose of this study, the OR represents the odds that an outcome (e.g. bankruptcy or non-bankruptcy) will occur given a particular exposure (e.g. debt ratio, firm size, liquidity position, profitability ratio etc.), compared to the odds of the outcome occurring in the absence of that exposure (Park, 2013).

The conditional probability $p = P\,(Y = Y'\,/X = x)$ is the probability that predicted the value of the dependent variable Y equals the interested outcome Y' given the values of the independent variables inputs x_i (Park, 2013).

The conditional probability $Pr(Y = y|X)$ is calculated by the following formula:

$$P_R\,(Y|X) = P(X)^Y * [1\text{-}P(X)]^{1-Y} \,, \quad (13)$$

Applying the LN operator both sides gives the following results:

$$LN[P_R\,(Y|X)] = Y*LN\,[P(X)]*(1\text{-}Y)*LN\,[1 - P(X)] \,, \quad (14)$$

The Log-Likelihood Function, LL, of Eq. (14) becomes

$LL=\sum_{i=1}^{m} Y_i * P(X_i) + (1- Y_i)*(1- P(X_i)$, (15)

In logistic binary regression, the objective of logistic regression is to find the coefficients of the Logit ($b_1 + b_1$ ... ... b_m) that maximize the function LL (i.e. the Log-Likelihood Function) (Park, 2013). The log-likelihood function is used to estimate the coefficients of the all the financial ratios used in the logistic binary regression (Park, 2013).

2.5.5.5 Hypothesis testing and variables selection

A common practice after fitting a multiple regression is to make specific decisions about the model (Park, 2013). These decisions include answering a question of whether all the predictor variables in the model are significant. Bremer (2012) states that a model with fewer predictors and about the same "explanatory power" is significant.

Testing for significance of variables is one of the major important steps for regression models (Park, 2013). This test usual asks whether any of the m predictor variables in the model have any relationship with the response where

$H_0 : b_1 = \cdot \cdot \cdot = b_m = 0$ vs. $H_\alpha : b_j \neq 0$ for at least one j. (16)

The test statistic function for this test is based on the sums of squares of the logistic regression mode (i.e. SS_R is the sum of squared residuals of the predicted and SS_R is the sum of squared residuals of the actual). The test statistic function can be written as follows:

$$F= \frac{SS_R /m}{SS_{Res} /(n-m-1)} , \quad (17)$$

where $SS_R=\sum_{i=1}^{m} (\hat{y}_i -\bar{y})* (\hat{y}_i -\bar{y})$, (18)

and $SS_{Res}= \sum_{i=1}^{m} (y_i -\hat{y}_i)* (y_i -\hat{y}_i)$, (19)

If the P value of the F-test statistic as shown in Eq. (17) is less than the significance level, the null-hypothesis is rejected. The conclusion is that the model provides a better fit than the intercept-only model (Park, 2013). The logistic regression model selects candidate variables using some form of ranking exercise called the Mean Decrease Impurity (MDI), the MDI concept is explained in detail in the next section.

2.5.5.6 Measure of variable importance Mean Decrease Impurity (MDI)

The dissertation did not use all the variables for the purpose of developing a logistic model. The final candidate variables were selected through a test of variable ranking called feature importance. Because of having too many variables it was statistically insignificant to implement an F- test statistic on all 72 variables. As a result, the variables were selected using a MDI (i.e. Mean decrease Impurity index) measure of the random forest technique.

In the context of random forest and ensemble randomize trees, Breiman (2002) proposed the measurement to evaluate the importance and ranking of variable X_m for predicting Y by adding the weighted impurity decreases $p(t)\Delta\, i(s_t,t)$ for all nodes t where X_m is used, averaged across all N_T trees in the forest (Louppe *et al*, 2013).

$$V_{(imp)}(X_m) = \frac{1}{N_T} \iint_t^T p(t)\Delta i(s_t,t) \quad , \tag{20}$$

where $t \epsilon T$ and $v(s_t) = X_m$ and where $p(t)$ is the proportion $\frac{N_T}{N}$ of samples reaching t and $v(s_t)$ is the variable used in split s_t (Louppe *et al*, 2013). The next section covers the application of machine learning techniques in Bankruptcy prediction including the different machine learning technique applicable in this study. Three different machine learning classification techniques are investigated.

3. Machine learning techniques in bankruptcy prediction

Machine learning is a field of computer science that gives computer systems the ability to learn with data and improve from experience without being explicitly programmed (Luxton, 2016). Bjorn (2017) states that since the early 1990s machine learning have been widely used for bankruptcy prediction. The machine learning techniques used in this study falls under both supervised and unsupervised learning.

Since learning involves an interaction between the learner and the environment, one can distinguish learning paradigms according to the nature of that interaction (Shai & Shai, 2014). Learning forms a very wide domain of machine learning. Machine learning is about extracting knowledge from data (Simon, 2009).

1. ***Supervised learning***, is where neural network is given the correct answers for a sample dataset and learn by example, an example of this type of learning can be illustrated by a task to detect spam emails. For the spam detection task, the learner receives training e-mails which are not labelled spam or No spam.

On the basis of such training the learner should create rules rule for labeling all newly incoming e-mail message as a spam or no spam based on the already received email, all the learner gets as training is a big dump of e-mail messages (with no labels) and the learner's task is to detect foreign messages (Shai & Shai, 2014). This research uses supervised learning as the model is given the correct answers for a sample dataset and learn by example.

2. ***Unsupervised learning***, is where the program is not given any examples and needs to use clustering techniques to analyse the given data. Simon (2009) explains that in unsupervised or self-organized learning, there is no external teacher or critic to oversee the learning process.

For a specific task, provision is made for which the network is required to learn from. Once the network has become tuned to proper weights of the input data, the network develops the ability to learn the task from the given inputs and thereby to create new classes automatically (Simon, 2009).

One of the biggest disadvantages of unsupervised learning is evaluating whether the algorithm learned something useful. Unsupervised learning algorithms are usually applied to unlabeled

data set. Therefore, it is very hard to say whether a model did well or not. As a consequence, unsupervised algorithms are more popular in exploratory assignment than automated tasks (Muller & Guido, 2017).

3. ***Reinforcement learning,*** is where the learning of an input–output mapping is performed through continued interaction with the environment in order to minimize the error and improve of performance the system (Simon, 2009). Outcome at the end is the reinforced learning, for example win/loose chess game.

Simon (2009) explains that the importance aspect of reinforcement learning is to minimize a cost-to-go function. The cost-to-go function is defined as the expectation of the cumulative cost of actions taken over a sequence of steps instead of simply the immediate cost Simon (2009).

Sometimes It may happen that certain actions taken earlier in that sequence of time steps are in the best determinants of the system behavior (Simon, 2009). The function of the learning system is to discover these actions and feed them back to the environment.

For the machine learning models, supervised learning was chosen, because the desired outcomes are given for the sample dataset and the models were trained to create classifier models by example. The machine learning techniques are explained in detail in the next sections, the focus will be on artificial neural networks, decision trees and support vector machines.

3.1 Artificial Neural Network

Artificial neural networks (ANN) commonly known as “neural network” were inspired by the behavior of biological neurons of the brain. The brain structure is a highly complex of neurons, with a nonlinear network (Simon, 2009). It possesses the power to organize its structural constituents, known as neurons in order to perform certain computations such as pattern recognition, perception, bicycle control etc. The computational efficiency of the brain is faster than the fastest digital computer in existence today (Simon, 2009).

ANN learns by examples and therefore is classified as supervised learning (Sanaz *et al*, 2016). Based on the learning error or experience, the weights used to predict the output are adjusted in order to progressively improve both the learning outcome and results. Continued interaction ensures that the neural network reduce the error of misclassification by updating the weight of the neural network structure (Simon, 2009).

Work on ANN has been motivated right from its inception by the discovery that the human brain functions differently from the usual digital computer. In the last few years, neural networks have received a significant of attention and were subject of research activity and practical applications across a range of fields (Al-Osaimy, 1998). Adya and Collopy (1998) have conducted a study to which analysed and compared neural networks with other different classification methods.

The above figure shows the different types of ANN classifiers (i.e. Binary or continuous value) and their corresponding type of learning methods (i.e. Supervised or Unsupervised). The supervised and unsupervised learning methods under binary inputs is further divided into different types of neural network memory systems in accordance to how each type of memory system processes information including the Hopfield network and Hamming network which falls under supervised learning, and the adaptive resonance theory which falls under the unsupervised learning of binary neural network classifiers.

A Hopfield network is a form of recurrent artificial neural network popularized by John Hopfield (1982). Since its development, one of the main uses of this network is in parallel content addressable memory in binary classification exercises. The units in Hopfield network are binary threshold units, usually takes values between 1 or -1. The output value is determined by whether or not the unit input exceeds their threshold or not (Hopfield, 1982).

Hamming network neural system are networks which establish commonalities between the input patterns and the weight vectors of all neurons through computing a distance and outputs the most active neuron selected by the so called MAXNET (Naoum *et al*, 2013). Hamming network uses computational element which are similar to those of the Hopfield (1982). Since its development, one of the main uses of this network is in parallel content addressable memory in binary classification exercises. The units in Hopfield network are

binary threshold units, usually takes values between positive one or minus one. The output value is determined by whether or not the unit input exceeds their threshold or not (Hopfield, 1982).

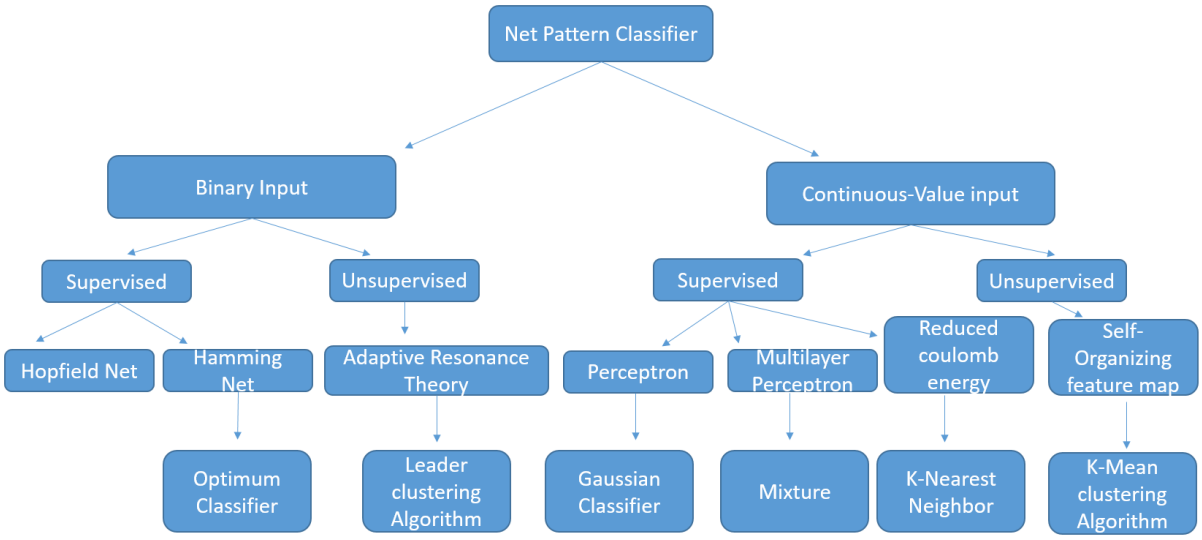


Figure 1: Summary of the different types of neural net classifiers corresponding to their classes (Frances, 2017)

An adaptive resonance theory (ART) is a theory a theory developed by Grossberg and Carpenter (1988). ART models incorporate new data by checking for similarities between this new data and known data. If there is a close enough match and or the new data resonates with what is in the memory, the new data is learned and the memory is adapted. If not, the new data is stored as a new memory, hence the name adaptive resonance theory.

McCulloch and Pitts (1943) describe the nervous system as a net of neurons having a soma and an axon where the adjunctions or synapses are always between the axon of one neuron and the soma of another neuron. This makes neurons as a logical operator in the processing of information in the human brain. A Neural network system comprise of computational units known as processing elements, proposed by McCulloch and Pitts (1943).

Because the ANN models were trained using supervised learning, a multi-layer feed-forward propagation artificial neural network model was chosen [7]. Each neuron has a propagation function s(x) which is the weighted sum of the inputs and weights associated with the input connections to the neuron. The propagation function is followed by an activation function which determines the output of the neuron.

The next section includes some of the past studies in bankruptcy prediction using neural and their evolution overtime.

3.1.1 Past studies on bankruptcy prediction

Evolution in computation technique have ushered a new era where Artificial Intelligence (AI) and machine learning form the backbone of bankruptcy prediction models (Naidu and Govinda, 2018). Amongst other machine learning prediction models, neural networks have also proven to be another efficient method for bankruptcy prediction.

Odom and Sharda (1990) developed a neural network model developed for prediction of bankruptcy. The model was tested using financial data from various companies. In testing the performance of the model, the same set of data was analyzed using traditional method of bankruptcy prediction, called the multivariate discriminant analysis. The difference between the current study and the study by Odom and Sharda (1990) is that the main focus on machine learning techniques was limited to neural network while the current study also explored decision trees and support vector machine.

Odom and Sharda (1990) compared the predictive abilities of both the neural network and the discriminant analysis. The results showed that neural networks were more applicable and efficient to the problem of bankruptcy prediction than the MDA. The current study compares all three machine learning models to each other first, and then compares all three of them to the traditional models instead of comparing one two types of methods as it was done in the study by Odom and Sharda (1990)

Naidu and Govinda (2018) proposed the use of neural network in bankruptcy prediction over traditional methods. Naidu and Govinda (2018) used a dataset based on polish companies, the study analysed 10503 companies from the year 2000 to 2012. Their prediction model was based around the analysis of financial ratios from the third year of forecasting period and the corresponding class label that indicates bankruptcy status after 3 years. This is different to the current study presented by this study as it forecast bankruptcy in the next 12 months based on past 12 months financial information.

The other difference was that Naidu and Govinda (2018) focused on only RF and ANN while the current study proposes both the RF and ANN including SVM for machine learning prediction methods. The study by Naidu and Govinda (2018) has proven that machine learning techniques like neural networks have proven to be more accurate than traditional approaches towards bankruptcy prediction.

Jardin (2010) evaluated the prediction accuracy of models designed using different classification methods depending on the technique used to select variables in bankruptcy prediction. He studied the relationship between the structure of the models and their ability to correctly predict financial failure. Another aim of the study was to investigate the influence of variable selection on the model performance.

There are similarities between the current study and the study by Jardin (2010) in the sense that both studies investigate the model design of the different classification methods. The only difference is that Jardin (2010) further investigate the influence of variable selection while the current study uses all variables except in the logistic model where variables are selected based on the feature importance method. The current study seeks to use the entire variable set in order to test the capability of machine learning techniques to correctly classify bankruptcy given a significant number of financial ratios.

Jardin (2010) mentioned that improving accuracy is definitely one of the most researched subjects in bankruptcy prediction. The aim is to assess the conditions under which a model performs well. This is usually done by optimising the number of factors and choosing the right parameter sets for the model. The author showed that neural network based model using a set of variables selected leads to better results than a set chosen with criteria used in the financial literature.

While some researchers have studied different methods of creating accurate models, other has focused on studying the variables to assess whether some predictors are better than others (Jardin, 2010). The investigation of variables by Jardin (2010) showed that variety of other factors, such as other samples, types of firms, sectors point to the need to use appropriate variable selection techniques to develop neural models (Jardin, 2010). However, the current study used all the variables but with the help of the feature importance characteristic the study ranks all the variables in terms of predictive power.

The next sections highlight the difference between the past studies and the current study.

3.1.2 Difference between past studies and current study.

The focus of the study is to investigate how different machine learning technique including ANN, Decision trees, and SVM can be used to predict bankruptcy in the management of credit risk within a financial institution. The study compares statistical regression model with various machine learning techniques in order to determine the best technique to use. This is different

to past studies which only looked at one method of bankruptcy prediction. The current studies also add to the existing debate which seeks to answer the best method of prediction between machine learning and statistical methods.

While statistical methods have remained the most common use of prediction in the past, there has been growing interest on machine learning techniques due to their ability to overcome some of the challenges that cannot be overcome by statistical methods such as regression. The next section therefore explains some of the benefits that comes with different machine learning techniques.

3.1.3 Benefits of ANN

An ANN derives its computing ability from its organized structure of neurons and its ability to learn from the inputs and therefore generalize. The process of *Generalization* in neural network refers to the neural production of reasonable outputs for inputs not encountered during training session (Simon, 2009).

In solving problems with big data sets, the following characteristic makes ANN a popular and capable tool to consider.

1. ***Nonlinearity***, an artificial neuron can be linear or nonlinear. A nonlinear neural network is one which is made up of an interconnection of nonlinear neurons. The Nonlinearity characteristic is very useful and important to solve nonlinear problems (Simon, 2009).

2. ***Input–Output Mapping***, an artificial neural network has the ability to adjust weights of the neurons (i.e. input) by applying a set of labeled training example. This is a popular method of learning called supervised learning. Supervised learning entails giving the correct answers for a sample dataset which then learns by example (Simon, 2009).

3. ***Adaptivity***, an artificial neural network has the power to adapt to new set of information (Simon, 2009). One limiting feature of automated techniques is their rigidity *(Shai & Shai, 2014).* Neural networks have the ability to be flexible and adjust their weights to changes in the surrounding environment. In particular, this ability makes them preferred to solve problems with changing assumptions (Simon, 2009).

3.1.4 Components of neural network

An artificial neural network scheme consists of simple processing units called the neurons. These neurons are connected using the weights where the strength of a connection between two neurons i and j is referred to as $w_{i,j}$ (Kriesel, 2007)

An ***artificial neural network*** is defined as a sorted triple $(\boldsymbol{N},\boldsymbol{V},w)$ with two sets $\boldsymbol{N}$, $\boldsymbol{V}$ and a function$w$, where $\boldsymbol{N}$ is the set of *neurons* and $\boldsymbol{V}$ a set $\{(i,j)|i, j \in \mathrm{N}\}$ whose elements are called connections between neuron i and neuron$j$. The function$w$: $V \rightarrow \mathbb{R}$ defines the weights, where $w((i,j))$, the weight of the connection between neuron i and neuron$j$, is shortened to $w_{i,j}$ (Kriesel, 2007).

Rosenblatt (1958) introduced the next major breakthrough in the study of artificial neural network system called the ***Perceptron***. Frances (2017) defines the perceptron as a function that represents the simplest form of a neural network used for the classification of linearly separable patterns. A perceptron consists of a single artificial neuron with adjustable weights and a bias (Frances, 2017). A perceptron is a linear neural network which is built around a single artificial neural network and only works in binary classification.

A perceptron algorithm calculates the weighted sum of the inputs using the following function:

$$\boldsymbol{Z}=\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{w_{i,j}.x_i} - \boldsymbol{\beta}\,, \tag{21}$$

Where:

$\boldsymbol{w_i}$ is the weight associated with the $\boldsymbol{i_{th}}$ input (attribute) $\boldsymbol{x_i}$

$\boldsymbol{\beta}$ is the bias term

$\boldsymbol{Z}$ is the perceptron potential

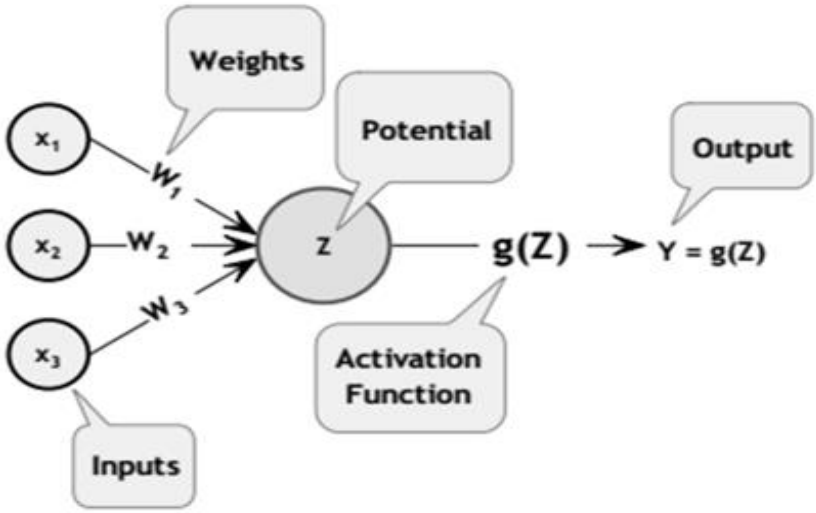


Figure 2: A Neural Net Perceptron (Frances, 2017),

Consider an example of a single layer feed forward neural network in figure 3:

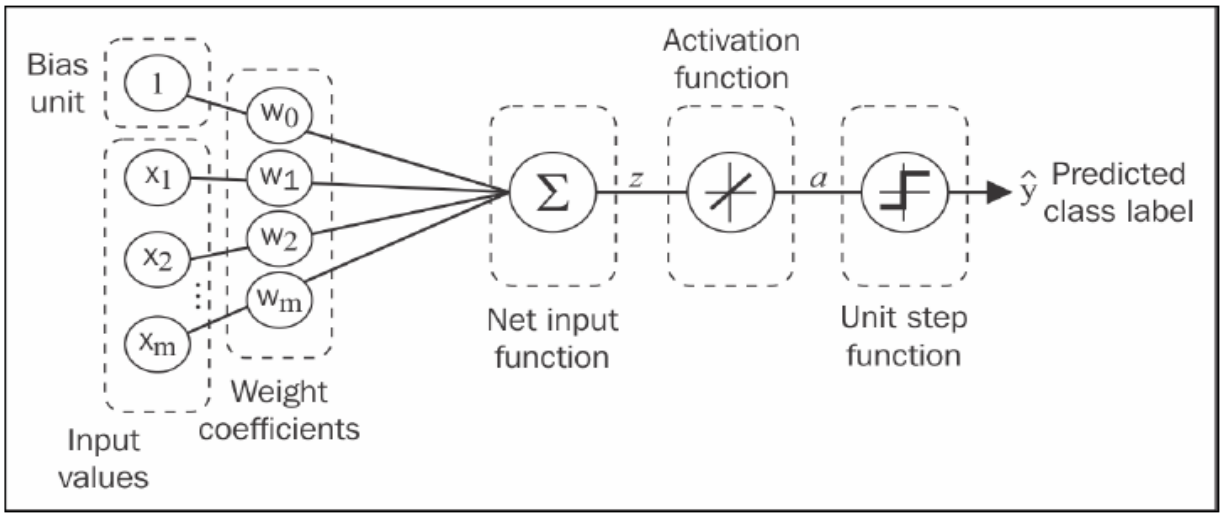


Figure 3: Consider an example of a single layer neural network (Raschka, 2016),

From figure 3 above, the values of the x_1,x_2, . . . , x_m represents the inputs, while the W_0,W_1, . . . , W_m represents the weights coefficients. The ***Net input function*** is also called the propagation function, while the **Unit step function** is the output which can be

3.1.4.1 Propagation function

A propagation function converts vector inputs to scalar network inputs for a neuron j .The propagation function receives the outputs x_{i1},x_2, . . . , x_{in} of other neurons i_1,i_2, . . . , i_n (which are connected to j) and transforms them in consideration of the connecting weights $w_{i,j}$ into

the network input net_j that can be further processed by the activation function(Kriesel, 2007). The network input is therefore the result of the propagation function (Kriesel, 2007).

The propagation function is sometimes referred to as the network input function (Rascka, 2016)

Let $I = \{i_1, i_2, \ldots, i_n\ n\}$ be the set of neurons, such that $\forall z \in \{1, \ldots, n\}$: $\exists$ $w_{i,zj}$. Then the network input of j, called net_j , is calculated by the propagation function f_{pro}as follows:

$$net_j = f_{pro}\ (o_{i1}. \ldots .o_{in,}. w_{i1}. \ldots .w_{in,j}) , \quad (22)$$

The weighted sum of the above is the multiplication of the output of each neuron i by $w_{i,j}$ gives the following equation:

$$net_j = \sum_{i=1}^{j} (o_i \ . w_{i,j}), \quad (23)$$

Kriesel (2007) states that before the training process begin, the set of weights need to be determined and assigned to the initial input values. Given a set of inputs$x_1, x_2, \ldots, x_m$ a weighted sum of the inputs is formed as follows:

$$C_j = \sum_{i=1}^{m} w_{i,j.} x_i + b_j \ , \quad (24)$$

where $w_{i,j}$is the network weights, x_i is the inputs and b_j the bias term. Here C_j is the activation and has the value of 1 if the neuron is activated and otherwise 0 (Kriesel, 2007). From the weight activation, the output in the next hidden layer of the neural network process is computed by applying the activation function using the weighted inputs of the previous layer (Kriesel, 2007).

Different activation functions including linear and non-linear functions have been proposed in the literature. Until recently, the most used activation functions have been the sigmoid and hyperbolic functions (Bjorn *et al*, 2017). In this study, the default rectified linear unit (ReLU) function was used from the Python scikit-learn library MLP function, however other optimiser functions such as the ELU, logistic, hyperbolic and the softmax are also investigated.

3.1.4.2 Activation function and activation term.

The activation function of a node defines the output given an input or set of inputs. The activation function is non-linear classifiers. Until recently, the most used activation functions

have been the sigmoid and he hyperbolic functions (Bjorn *et al*, 2017). ANN's are the first machine learning technique applied in the bankruptcy prediction problem (Goeletsis *et al*, 2012), It is also considered the most studied one (Sanaz *et al*, 2016).

The activation function determines the activation of a neuron which is dependent on the network input and threshold value. The activation a_j of a neuron j depends on the previous activation state of the neuron and the external input (Kriesel, 2007).

Activation functions are functions used in neural networks to computes the weighted sum of the inputs and errors which are commonly referred to as biases (Nwankpa *et al*, 2018). The biases are used to decide if a neuron can be eliminated or not (Nwankpa *et al*, 2018). Activation functions manipulate the training data in order to produce an output for the neural network through some gradient processing. The most common used gradient method is usually the gradient descent (Nwankpa *et al*, 2018).

Kriesel (2007) defines the mathematical relationship of activation function and activation term as follows: Let j be a neuron and Ω be the threshold value. The activation function is defined as:

$$a_j\,(t) = f_{act}\,(net_j\,(t), a_j\,(t\,-\,1), \Omega_j), \tag{25}$$

3.1.5 Common activation functions

Unlike the other variables within the neural network, the activation function is usually defined once and globally applied (Kriesel, 2007). Only the threshold values are different for each neuron in the network system (Kriesel, 2007). Threshold values can be changed by introducing a new set of information. So it is very important to relate the threshold value to the time and to write it as $\Omega_j \rightarrow \ \Omega_j(\mathrm{t})$ (Kriesel, 2007).

Deep neural networks have been successfully used in various sectors to solve real world complex problems (Nwankpa *et al*, 2018). Activation functions are thus required to achieve accurate predictions by performing diverse computations between the hidden layers and the output layers of any given deep learning architecture (Kriesel, 2007).

3.1.5.1 Sigmoid Function

The Sigmoid activation is commonly referred to as the logistic function or squashing function in some literature (Nwankpa *et al*, 2018). The Sigmoid is a non-linear activation used most often in feed-forward neural networks for predicting probability based output.

The Sigmoid activation function has been successfully applied in binary classification problems, modeling logistic regression tasks as well as other neural network domains (Neil, 1992). However, the disadvantage of sigmoid functions is the fact that the models hardly learn something for values far from the threshold value, unless the network is modified (Kriesel, 2007).

The Sigmoid function is given by the following relationship: The sigmoid function is used for (0,1) models.

$$f(x) = \frac{1}{1+exp^{-x}} \ , \tag{26}$$

3.1.5.2 Hyperbolic tangent function (Tanh)

The hyperbolic tangent function is another type of activation functions used in deep learning. The tanh function is used for (-1,1) models. Verbi and Karklik (2011) argues that the tanh function is the most preferred function compared to the sigmoid function because it gives better training performance for multi-layer neural networks (Verbi & Karklik, 2011). However, the choice between the sigmoid and the than depends on the output model chosen. The disadvantage of the hyperbolic function is that it is not a gradient descent problem solving method when compared to the sigmoid. However, the main advantage provided by the function is that it produces zero centered output which boosts the performance of the back-propagation process (Nwankpa *et al*, 2018).

The hyperbolic tangent function known as tanh(x) is a function whose range lies between -1 to1 (Nwankpa *et al*, 2018). The output of the tanh function is given by

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad , \tag{27}$$

3.1.5.3 Softmax function

The Softmax activation function is commonly used in multi-class prediction models where the output is the probability of each class such that the target class is the one having the highest probability (Nwankpa *et al*, 2018). Krizhevsky (2012) has successfully used the softmax

activation function in deep convolutional neural network to classify the 1.2 million high resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. The main difference between the Sigmoid and Softmax activation function is that the Sigmoid is used in binary classification while the Softmax is used for multivariate classification tasks.

The Softmax function is computed using the relationship:

$f(\mathrm{x_i}) = \frac{\exp(\mathrm{x_i})}{\sum \exp(\mathrm{x_i})}$, where x_i is the input values (28)

3.1.5.4 Rectified linear unit (ReLU) Function

The ReLU function was first proposed by Nair & Hinton (2010) and has since become the most widely used activation function for deep learning applications (Nwankpa *et al*, 2018). The ReLU is a faster learning activation function and has proved to be the most common use (Ramachandran *et al*, 2017). Zeiler *et al* (2013) states that the ReLU can improve generalization and make training of deep networks faster and simpler by substituting the logistic units with rectified linear units.

The ReLU activation function offers a better performance and generalization in deep learning compared to the Sigmoid and tanh activation functions (Dahl *et al*, 2013). The ReLU activation function performs a threshold operation to each input element of the neural network system where values less than zero are set to zero (Nwankpa *et al*, 2018).

The ReLU is given by the following mathematical relationship:

$$f(x) = max(0, x) = x_i \; if \; x_i \geq 0 \; OR \; 0 \; if \; x_i < 0, \quad (29)$$

The above function rectifies the values of the inputs less than zero thereby forcing them to zero (Nwankpa *et al*, 2018). This elimination property of the ReLU activation function ensures that the gradient problem observed in the earlier types of activation function is not a requirement or a necessity (Nwankpa *et al*, 2018).

The ReLU function has been successfully used within the hidden units of the deep learning neural network with another activation functions for speech recognition purposes (Nwankpa *et al*, 2018). Zhang *et al* (2015) successfully used the ReLU activation function in neural networks for image classification and the results surpassed human-level performance on the visual recognition challenge

Maas and Hannun (2013) used the ReLU in hidden units to demonstrate additional gains in final system performance compared to a sigmoid activation function in deep neural network acoustic models used for speech recognition. The main advantage of using the ReLU activation function in computational assignment is that it guarantees faster computation since it does not compute any heavy calculations such as exponentials and divisions (Nwankpa *et al*, 2018).

Different layers can use different activation functions, according to Raschka (2016), the most applicable activation function for classification problems is the ReLU. The ReLU is the default activation function in python classification models. The study tests all the above activation functions as they are applicable for binary classification except the Softmax, the Softmax is used for multivariate classification tasks. Raschka (2016) stated that the maximum number of layers that can be used is five layers.

3.1.5.5 Exponential linear unit function

The exponential linear units (ELUs) are commonly used to speed up training of deep neural network and were first proposed by Clevert *et al* (2015). The main advantage of the ELUs is that they can eliminate the diminishing gradient challenge by using identity for positive values and also improves the learning characteristic, the ELU represents a good alternative to the ReLU as it decreases bias (Nwankpa *et al*, 2018).

The advantages of ANN are as follows (Frances, 2017).

1. ANN algorithms are easy to implement.
2. ANN's are good for approximating complex nonlinear mappings.
3. ANN's parameters are easy to adjust and use.
4. Their applicability is ranges across wide range of problems.

The disadvantages of ANN's are as follows (Frances, 2017).

1. ANN's lack transparency and work like a black box system.
2. Not all problems can be solved using the same number of neurons and layers, the parameter optimisation remains the biggest challenge in the ANN prediction exercises.

3.2 Random forest

An ensemble of classifiers is a set of classifiers whose individual decisions are combined to classify new set of information using the weight of the prediction (Dietterich,2000). Random

forests are an ensemble learning method used for classification, the methodology includes construction of decision trees of the given development data and matching the validation data with these (Zoubin, 2012). The most popular ensemble variant of the decision tree is named the *Random Forest*. It is commonly known for being less prone to overfitting (Wagemans. 2017). Random forests are used to rank the importance of variables in a classification problem

Decision trees techniques are another class of machine learning algorithms which are widely used in both regression and classification problems (Simon, 2009). Generally, decision trees use mathematical formulations in order to detect the best possible combination of important predictors and create a tree structure for deriving the classification decisions (Goeletsis *et al*, 2012). In order to reach a decision, the model keep answering the questions until it reaches a terminal node. Decision trees are popular models for classification and regression tasks. They learn a hierarchy of if/else questions, leading to a decision (Simon, 2009).

The decision tree model learns a series of questions to infer the class labels of the samples. The intermediate nodes are often referred to as internal nodes and terminal nodes are called leaves (Bjorn *et al*, 2017). Based on the answers for the question posed the model predict the most appropriate activity for the day. Decision trees can detect relationships by means of independently learning the interaction of input variables and also disregards redundant input variables (Candice, 2016).

The goal with a decision tree in most cases is to find the best split between input variables (Candice, 2016). The structure of the tree can be better explained by splitting the input variable into branches with nodes, the decision tree model learns a series of questions to infer the class labels of the samples into the right node (Bjorn *et al*, 2017). The intermediate nodes are often referred to as internal nodes and terminal nodes are called leaves (Bjorn *et al*, 2017).

Sharma and Sunil (2016) explained a decision tree scheme as a flowchart-like tree structure, where each internal node represents a test on an attribute. Each branch represents an outcome of the test of each node. Given a tuple X of different attributes, the attribute values of the tuple are tested against the decision tree. Each model is defined as a path which can be traced from the root to a leaf node which holds the classification of each set of tuples.

Huttunen (2015) describe a decision tree as a straight forward if-then-else-like diagram that combines individual attributes into a decision. Random forest the most popular tree based classifier proposed by Breiman in 1993. The random forest is based on a collection of decision

trees. Just like the supervised neural network system, decision trees are easily trained to learn the data, therefore they follow a supervised learning paradigm.

A decision tree is a prediction function, f {X → Y}, that predicts the label associated with an instance x by traveling from a root node of a tree to a leaf (Shai & Shai, 2014). For simplicity and for the purpose of this study, the focus is on the binary classification setting, where Y = {0, 1}. Most decision tree classifiers are designed to classify the data with categorical or Boolean class labels (Patel &Singh, 2015). Decision trees are not limited to binary classification.

At each node on the root-to-leaf path, the successor child is chosen on the basis of a splitting of the input factor. The splitting is based on one of the features or characteristic of x on a predefined set of splitting rules, where x represents the measured attributes. Usually, each node in the tree either represents a question or a terminal node (also called a leaf) that contains the answer. The edges connect the answers to a question with the next question to be asked.

One of the biggest challenges of decision trees is they overlearn the data, i.e. training data is memorized with poor ability to generalize (Huttunen, 2015). With no predefined splitting rules, decision trees have each root to leaf structure which is similar to its nearest neighbor. A common splitting rule at internal nodes of the tree is based on assigning a tolerance value on each single feature, a split to the right or left child of the node on the basis of [x_i<μ], where i Ɛ [d] is the index of the relevant feature and μ Ɛ ℝ is the threshold (Shai & Shai, 2014).

To avoid this weakness, the decision tree is trained using many trees where each tree is trained with a subset sample of training data. To avoid overfitting, some of the attributes are hidden from the training.

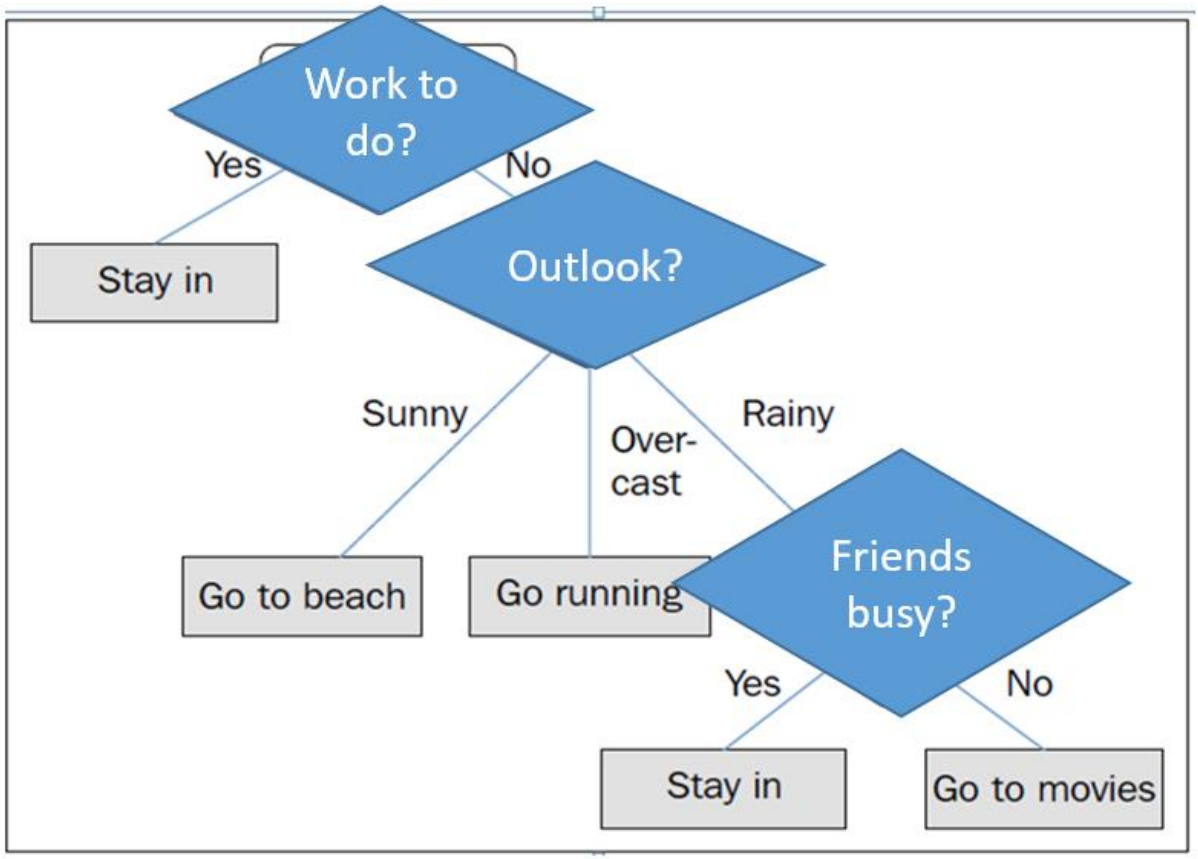


Figure 4: Consider an example of a decision on an activity using on a particular day using decision tree (Raschka, 2016)

The above figure illustrates an example where a decision tree can be used to decide upon an activity on a particular day. Based on the features in our training set, the decision tree model learns a series of questions to infer the class labels of the samples. The starting node is asking if someone has work to do or not. Based on the binary response, each question lends on a different split, where a "Yes" means that someone will stay in and do the work and a "No" leads to a follow up question about the weather outlook.

The response on the weather outlook will result in three split, a sunny weather will mean that someone will go to the beach, an overcast will result in going to run while a rainy weather will lead to a further split based on the schedule and availability of friends. If Friends a busy and the weather is rainy, the decision will be to stay indoors. However, If the weather is rainy and the friends are not busy, then a decision to go to the movies will be taken.

3.2.1 Measuring best split using entropy

At each question node a splitting rule is used. For example, in figure 4, the root node can be either a "YES" or "NO". Different criteria can be used to determine the best split available. The most used criterion is to choose the split that maximizes the information gain (Bjorn *et al*, 2017). The information gain is a measure of the reduction of uncertainty after a split. To measure uncertainty, entropy is used. Formally the entropy of a state *s* is defined as:

$$E(s) = -\sum_{i=1}^{m} p(i)\log(p(i)), \quad (30)$$

where m is the number of branches and $p\ (i)$ the probability of each branch i

When the main problem is to classify bankrupt firms from non-bankrupt firms, the decision tree creates an optimum tree structure of the predictors, which map to a specified outcome branch. In this case the prediction of bankruptcy or not in the next 12 months is the outcome. Marais *et al* (1984) used decision trees by investigating an application of recursive partitioning and bootstrapping to commercial bank loan classifications, Frydman *et al* (1985) states that decision trees are more superior to statistical methods in predicting corporate failure in their case studies.

The information gain of example set *S* on attribute *A* is defined as

$$Gain(S, A) = E(s) - \left(\sum_{i=1}^{m} \frac{|S_V|}{(|S|)} E(S_V)\right), \quad (31)$$

where S_V is a subset of *S* for which attribute *A* has value v, and $S_1, \ldots . S_m$ are the partitions induced by attribute A . The attribute value that maximizes the information gain is chosen as the splitting attribute (Ning *et al*, 2007)

3.2.2 Variable selection using feature importance and random forest

One of the most important issues in the bankruptcy prediction problem is to employ an effective variable selection algorithm which will lead to an acceptable level of performance accuracy in the implementation stage. Applications of combined variable selection methods and classification models including machine learning techniques are topics that have been given the most attention in recent years (Louppe *et al*, 2013).

Applications of combined variable selection methods and classification models including machine learning techniques are topics that have been given the most attention in recent years (Fallahpour *et al*, 2017). For the purpose of this study, the predictive variables are chosen based

on the same criteria. This is done in Python using the random forest "feature importance "built in function.

Random forest is a supervised learning method that builds an ensemble of t decision trees (Wehenkel *et al*, 2018). When predictor variables are numerical in nature, a decision tree is a dichotomous tree where each node is given a weight based on the calculated mean. The means of the inputs are compared to a threshold value and where each leaf node is labelled with a prediction of the output class of either a 0 or 1.

Random forests are a combination of tree predictors proposed by Leo Breiman in the 2000's (Biau, 2012). Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest (Breiman, 2001). Despite growing interest and practical use, there has been little exploration on the statistical properties of random forests and as a result little information is known about the mathematical forces driving the algorithm (Louppe *et al*, 2013).

Breiman (2001) promoted random forests for use in classification tasks. His study suggested using averaging as a means of obtaining good discrimination rules. The classifiers used for averaging are randomized (Biau, 2012). This algorithm assigns the highest probability to a variable with the most predictive power and a lower variable to the variable with the least predictive power (Raschka, 2016).

The algorithm has since been successfully applied in many areas (Kartasheva & Traskin, 2011). These sections provide a simple example which illustrates how the Random Forest algorithm works. The study considers a small sample of eight companies which contains two groups of companies (i.e. Bankrupt and non-bankrupt). Each company has two characteristics, x and y, with $x;\ y\ \epsilon$ [0; 1]. Using binary classification where zero represents non-bankrupt and one represents bankruptcy.

Kartasheva and Traskin (2011) mathematically define all companies to be located on a square matrix U = [0; 1] *[0; 1].

The classification goal is to divide the square U into two subsets S and N, such that $S \cup N = U$. The classification must be such that a company i is classified as bankrupt if $(x_{i,} y_i) \epsilon S$ and non-bankrupt if $(x_{i,} y_i) \epsilon N$

Kartasheva and Traskin (2011) mentioned that the Random Forest has many other several characteristics which make it suitable for bankruptcy prediction compared to other methods,

one which is its ability to adapt to non-linear decision boundaries between bankrupt and non-bankrupt classes. Random forest does not make any restrictive model assumptions on the relationship between the dependent and independent variables and therefore makes it one of the highly flexible procedures for classification exercises (Kartasheva and Traskin, 2011).

3.3 Support vector machines

Support vector machines (SVM) are kernel based support vector regression (SVR) methods, this is a supervised learning algorithm where correct answers are given for a sample dataset and learn to classify by example. These methods are based on mapping data from the original input feature space to kernel feature space of higher dimension to a linear problem in this space. According to Cortes and Vapnick (1995), the goal of the SVM is to produce a model which predicts target values of data instances in the testing set in which only the input predictors are given.

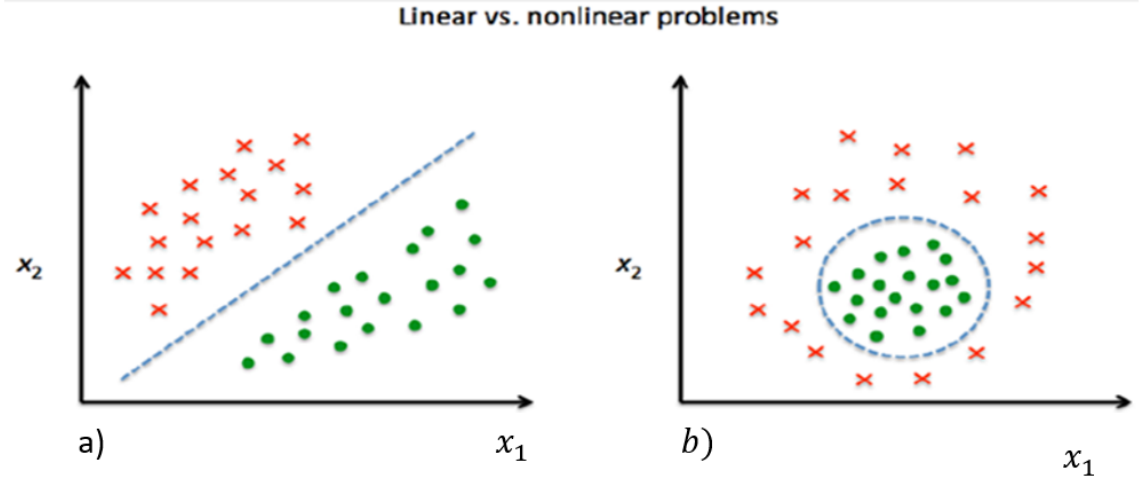


Figure 5: An example of a) linear and b) non-linear separable problem (Raschka, 2016)

The solution of a SVM based problem is binary in nature. Figure a) above is linearly separable as there exist a line which linearly separates the red crosses from the green dots as seen above. However not all problems are linearly separable, figure b) shows a case where the red crosses cannot be linearly separated from the green dots. This is an example of non-linear separable problem. When the problem is non-linear, the SVM maps the original data from an original space into a feature space using a kernel function then solves a linear problem in that space.

3.3.1 SVM optimisation problem

Let the training labelled data set of pairs be represented by$(x_{i,}y_i)$ with i =1…m, where x_i E $\mathbb{R}^n$ is the training vector belonging to one of the binary classes. m represents the total number of extracted features or the independent variables in the training data set and y_i represents the class which x_i belongs to.

The support vector machine requires a solution for the following optimisation problem:

$$min\ _{a,b,\xi}= (\ \frac{1}{2}W^T\mathrm{W} + \mathrm{C}\sum_{i=1}^{m}\ \xi_i\ ), \tag{32}$$

Subject to

$$y_i\ (\boldsymbol{W}^T\ \Psi(x_i\ ) + b \geq 1 - \xi_i\ , \xi_i\ \geq 0) \text{ for all } y_i \tag{33}$$

From the conditional equation above, b represents the bias term. **W** is a vector perpendicular to the hyperplane <**w**, *b*>, ξ is the factor of classification error and *C*>0 is the penalty parameter of the error term (Goeletsis *et al*, 2012).

In SVM, the training vectors x_i are mapped into a higher dimensional feature space F by the functionΨ: $\mathbb{R}^n\rightarrow$ F. Support vector machines find a separating hyperplane with the maximal geometric margin and minimal empirical risk $R_{\ emp}$ in the higher dimensional space (Goeletsis *et al*, 2012). $R_{\ emp}$ is defined by the following equation:

$$R_{\ emp}(e) = \frac{1}{2m}\sum_{i=1}^{m}\ \left|y_i\ \text{-}f(x_i\ , e)\ \right|, \tag{34}$$

From Eq. (34) above, f represents a decision function defines as follows:

$$f(x\ ) = \sum_{i=1}^{m}\ y_i\ e_i\ \mathrm{K}\ (x_i\ , x) + b\ , \tag{35}$$

Where K $(x_i\ , x_j\ ) \equiv \Psi\ (x_i\ )^T\ \Psi\ (x_j\ )$ represents the kernel function. b represents the bias term and e_i are the weights.

The binary threshold units, usually takes values between 1 or -1. The output value is determined by whether or not the unit input exceeds their threshold or not (Hopfield, 1982).

This study uses the radial basis function (RBF) as the basic kernel of SVM because it usually gets better results than other kernel functions (Min & Lee, 2005). The RBF kernel, also called the Gaussian kernel, is a kernel that is in the form of a Gaussian function (Matthew, 2017).

A RBF kernel which is defined as follows is used:

$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, with $\gamma > 0$, (36)

and

$$\gamma = \frac{1}{2\sigma^2}\ , \quad (37)$$

From Eq. (37) above, σ is a kernel parameter which represents the standard deviation (Matthew, 2017).

3.3.2 Feature space mapping

Let $T = \{(x_i, y_i)\}$; $i = 1 \ldots k$; $x_i \ E \ \mathbb{R}^n$; $y_i \in \{-1, +1\}$ a linear separable training set. Then there exists a hyperplane of the following form:

$$\bar{W}^T x_i + b = 0, \quad (38)$$

Separating the positive from the negative training examples such that

$$\bar{W}^T x_i + b \geq 0 \text{ or } y_i = +1, \quad (39)$$

$$\bar{W}^T x_i + b < 0 \text{ or } y_i = -1, \quad (40)$$

where $\bar{W}$ is the normal to the hyperplane and b is the perpendicular distance of the hyperplane to the origin (Hofmann, 2006).

Consider an example non-linear mapping function below:

Ω: $I = \mathbb{R}^2 \rightarrow F = \mathbb{R}^3$ from the 2-dimensional input space I into the 3-dimensional feature space F, which is defined in the following way

$$\Omega(\bar{x}) = (x_1^2\ , \sqrt{2x_1}\, x_2\ , x_3^2)^T\ )\ , \quad (41)$$

Taking equation for a separating hyperplane Eq. (41) into account, the result is a linear function in $\mathbb{R}^3$:

$$\bar{W}^T \Omega(\bar{x}) = w_1 x_1^2 + w_2 \sqrt{2x_1}\, x_2 + w_3 x_3^2\ , \quad (42)$$

After mapping a non-linear separable data into a higher dimensional space, it deduces to a linear separating hyperplane as shown by Eq. (42) above (Hofmann, 2006).

The kernel non-linearly maps training samples into a higher dimensional feature space in order to handle non-linear relationship between class labels and attributes (Hofmann, 2006). The SVM achieves this by constructing a linear model to estimate the decision function using linear

or non-linear class boundaries (Hofmann, 2006). The main aim of a Support Vector Machine is to devise a computationally efficient way of learning good separating hyperplanes in a high dimensional feature space (Hofmann, 2006)

As illustrated in figure 6, the kernel trick method can be used to make data which is non-linear separable to a linear separable data in a feature space. Since the linear machine can only classify the data in a linear separable feature space, the role of the kernel-function is to induce such a feature space by implicitly mapping the training data into a higher dimensional space where the data is linear separable (Hofmann, 2006). Kernel principal components analysis uses the eigenvectors and eigenvalues of the data to draw conclusions from the directions of maximum variance to construct inner product kernels (Hofmann, 2006).

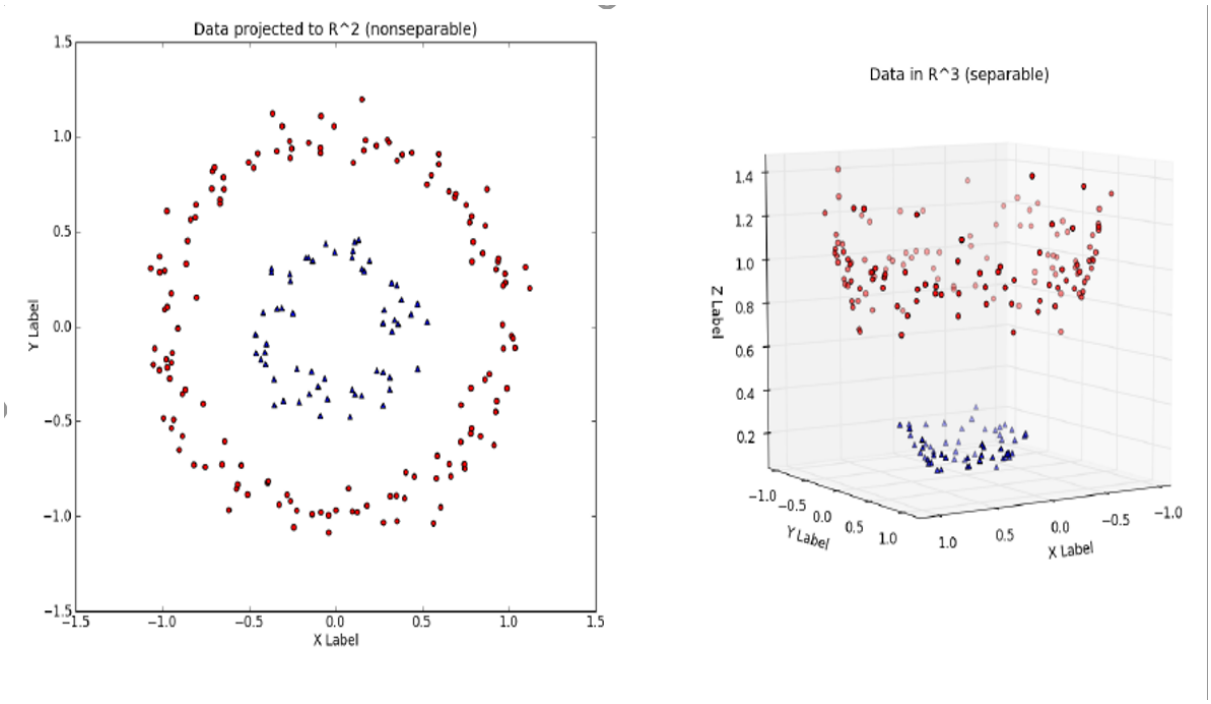


Figure 6: An example of transforming non-linearly separable data to linearly separable data (Kim, 2013)

3.3.3 Different types of kernels and their properties

The most important question in SVM is how to find the most appropriate kernel (Hofmann, 2006). The study has already discussed so far the functionality of kernel functions and their use with support vector machines. Hofmann (2006) defines the kernel function as a kind of similarity measure between the input objects. Eq. (35) above has already defined the RBF kernel which is used in this study. Table 1 explains other different forms of kernel functions that exist.

Table 1: A summary defining different types of common product kernels (Haykin, 1998)

Type of Kernel	Inner product kernel $K(\vec{x}, \vec{x}_i),\ i = 1, 2, \ldots, N$	Comments
Polynomial Kernel	$K(\vec{x}, \vec{x}_i) = \left(\vec{x}^T \vec{x}_i + \theta\right)^d$	Power p and threshold θ is specified a priori by the user
Gaussian Kernel	$K(\vec{x}, \vec{x}_i) = e^{-\frac{1}{2\sigma^2}\|\vec{x}-\vec{x}_i\|^2}$	Width σ^2 is specified a priori by the user
Sigmoid Kernel	$K(\vec{x}, \vec{x}_i) = tanh(\eta\, \vec{x}\, \vec{x}_i + \theta)$	Mercer's Theorem is satisfied only for some values of η and θ
Kernels for Sets	$K(\chi, \chi') = \sum_{i=1}^{N_\chi} \sum_{j=1}^{N_{\chi'}} k(x_i, x'_j)$	Where $k(x_i, x'_j)$ is a kernel on elements in the sets χ, χ'
Spectrum Kernel for strings	count number of substrings in common	It is a kernel, since it is a dot product between vectors of indicators of all the substrings.

For the purpose of this study where the problem is corporate failure or credit default prediction, the decision hyperplane produces binary classes, which maps to a specified outcome class. In this study the outcome is a binary classification which determines bankruptcy or not using support vectors. The decision model used to separate the two classes is called an optimal separating hyperplane. In prior studies, the SVM has been proven to outperform other non-linear techniques including ANNs based non-linear prediction techniques (Jayanthi *et al*, 2011).

Advantages of SVM's are as follows (Frances, 2017):

1. The introduction of kernel makes SVM gains flexibility while separating one class from another
2. The kernel inherently contains a non-linear transformation. The transformation happens implicitly without the help of human knowledge
3. SVM has good generalization ability even when the modeling data is biased.

Disadvantages of SVM's are as follows (Frances, 2017):

1. The input variables remain independent variables even though the polynomial kernel attempts to model the interaction among the variables
2. SVM works efficiently on data with few missing values
3. No straight forward way on how to choose the optimal input feature subset for
4. The computational cost is high when there is large amount of features
5. The only way to explain SVM results is only graphically since they do not assign any parametric score function
6. Weights of independent variables are not constant.

3.4 Hyper-parameter tuning

In order to know which model performs well, machine learning methods such as neural network, support vector machine and decision trees entails using both appropriate cross validation and right parameter tuning for good prediction results. Different cross-validation techniques can be used where the training dataset is further divided into training and validation subsets in order to estimate the generalization performance of the model (Raschka, 2016).

In the past, hyper-parameter tuning has been the job of humans because they can be very efficient in solving problems where only a few trials are possible (Bergstra *et al*, 2011). However, computational algorithms have become very effective on running more trials in a very short space of time than humans (Bergstra *et al*, 2011). The authors had conducted a study to show that algorithmic approaches can find better results in less time than humans.

Raschka (2016) states that one cannot expect that the default parameters of the different learning algorithms provided by software libraries are optimal for a specific problem task. The results of the final selected machine learning models were achieved through the use of hyper-parameter optimisation techniques that help to fine-tune the performance of algorithm. After the selected model has been fitted on the training dataset, the test dataset is used to estimate how well it performs on this unseen data to estimate the generalization error.

3.5 Grid search

Frances (2017) defines the grid search method as a Brute Force Method for optimisation. It is a science to parameter tuning which build and evaluate a model for each combination of parameters specified in a grid of the algorithm. This methodology allows the assessment of different combinations of hyper-parameters and finds the most accurate combination with the highest prediction rate. The parameters are usually specified as a vector of possible values defining all possible combination in the grid search (Frances, 2017).

3.6 Cross validation

Once the best model has been chosen based on the approach of a scientific grid search, the model needs to be tested on new data that has never been seen before. Generally, if there was enough data, it is recommended to set aside a validation set and uses this to assess the performance of the model (Hastie *et al*, 2009). After the validation set has been set aside, the training and testing set is split using the 70/30 respectively.

In classification and modelling using machine learning algorithms, the process of splitting the data set into modelling and testing data set is known as cross-validation (Frances, 2017). Refaeilzadeh *et al* (2009) defines Cross-Validation as a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model.

3.6.1 Holdout -cross validation

Frances (2017) explains one of the simplest most used methods of testing called the holdout method. The holdout method, sometimes called test sample estimation, splits the data into two mutually exclusive subsets (i.e. training set and a test set). In this type of validation approach, it is common to split the data by assigning two third of the data as the training set and the remaining one third as the test or household set.

3.6.2 K-fold cross validation

Kohavi (1995) explains that the disadvantage of using the house hold sample is that one third of the modelling data is not used to train the model. As a results of this challenge, the K-fold cross validation exists as a solution to improve the hold-out methodology. In this type of approach, the dataset is divided into k-folds subsets. Then one of the k subsets or folds is used

as the test set whiles the remaining k- 1 subsets are combined to form a training set which the model is built upon.

3.6.3 Leave-one-out cross-validation

The Leave-one-out cross-validation, generally known as the (LOOCV) is another method of cross validation where the number of f-folds subsets equals the number of instances N in the data set. Frances (2017) explains that according to Efron (1982), the function approximation is trained on all the data except for one point and a prediction is made for that excluded point.

3.7 Grid search and cross-validation optimisation in machine learning

When using machine learning algorithms to perform classification exercises, the grid search method is nested within the cross-validation loop to select the optimal parameters for each algorithm (Frances, 2017). A grid was created using different parameters depending on whether the classification is for an artificial neural network, support vector machine or decision tree algorithm.

3.8 Feature selection and variable importance

The pioneer's studies on variable and feature selection in machine learning started as early as 1997 (Blum and Langley, Kohavi and John, 1997). Variable importance and feature selection have become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available (Guyon, 2003).

Several methods use simple correlation coefficients similar to Fisher's discriminant criterion while others adopt mutual information or statistical tests (t-test, F-test) (Ladha and Deepa, 2011). Frances (2017) explain that there are three general approaches to solve the feature selection problem, namely the wrapper method, filter method and embedded method.

Fallahpour *et al* (2017) conducted a study and introduced a precise feature selection model and compared the obtained results with those of other conventional models without feature selection but tackling bankruptcy prediction in terms of accuracy. The proposed method involved the using the sequential floating forward selection (SFFS) algorithm applied as a wrapper feature selection technique to determine the best subset of features (Fallahpour *et al*, 2017)

The wrapper method uses the learning machine of interest as a black box to score subsets of variables according to their predictive power (Guyon, 2003). Filters select subsets of variables as a pre-processing step, independently of the chosen predictor whereas Embedded methods perform variable selection in the process of training and are usually specific to given learning machines (Guyon, 2003).

The wrapper method offers a simple and powerful way to address the problem of variable selection, regardless of the chosen learning machine method. However, wrapper is criticized for the following weakness. They seem to be using a "brute force" method requiring massive amounts of computation (Guyon, 2003). Embedded methods are commonly used in decision trees such as CART. According to Guyon (2003), CART has a built-in mechanism to perform variable selection.

Guyon (2003) compared the wrapper method and filter method, the author argued that filters are generally faster. Another argument some filters which is based on mutual information criteria tends to provide a generic selection of variables which is not tuned for a given learning machine. The author continued to demonstrate how filters are known for their great ability to reduce space dimensionality while at the same time overcome overfitting. These methods select features based on discriminating criteria that are relatively independent of classification (Ladha and Deepa, 2011).

Kohavi and John (1997) classify the variable ranking assignment as a filter method because it is a preprocessing step, independent of the choice of the predictor. Several methods use simple correlation coefficients similar to Fisher's discriminant criterion filter methods are usually recommended for their fast data analysis ability (Ladha and Deepa, 2011).

However, it is argued that in order to efficiently validate the results and optimize the process of input variables selection the wrapper method is the most efficient (Frances, 2017). Kumari and Swarnkar (2011) argued that the wrapper method is the most efficient in selecting the most predictive variables for building a classification model.

3.9 Steps to determine feature selection problem

Guyon (2018) provides the following steps for feature selection

1. **Is there domain knowledge?** If yes, construct a better set of "ad hoc" features.
2. **Are there features commensurate?** If no, consider normalizing them.

3. **Does suspicion of interdependence of features?** If yes, expand your feature set by constructing conjunctive features or products of features

4. **Is there a need to prune the input variables** (e.g. for cost, speed or data understanding reasons)? If no, construct disjunctive features or weighted sums of features (e.g. by clustering or matrix factorization)

5. **Is there a need for assessment of features individually** (e.g. to understand their influence on the system or because their number is so large that there is a need to do a first filtering)? If yes, use a variable ranking method else, do it anyway to get baseline results

6. **Is a predictor needed?** If no, stop

7. **Is there any suspicion that the data is "dirty"** (have a few meaningless input patterns and/or noisy outputs or wrong class labels)? If yes, detect the outlier examples using the top ranking variables obtained in step 5 as representation; check and/or discard them

8. **Is the data to be first tested known**? If no, use a linear predictor.3 Use a forward selection method with the "probe" method as a stopping criterion

9. **Are there new ideas, time, computational resources, and enough examples**? If yes, compare several feature selection methods, including new idea, correlation coefficients, backward selection and embedded methods

10. **Is there a need for a stable solution** (to improve performance and/or understanding)? If yes, subsample the data and redo the analysis for several bootstraps.

According to Ladha and Deepa (2011), the following are two general approaches in feature selection:

1. **Forward Selection**: Begin the process with no input and add them one by one, at each step adding the one that decreases the error the most, until further additions makes no material difference

2. **Backward Selection**: Includes all variables and remove them one by one, at each step removing the one that decreases the error the most until further removals makes no significant changes.

3.10 Types of search method for variable selection

There are three different methods (i.e. wrapper, embedded and filter methods) used in variable selection type.

3.10.1 The filter method

The filter method creates an array of elements of the original array that passed the set test. All elements which did not pass the set test are filtered out. The filter method is split into univariate and multivariate type (Ladha and Deepa, 2011)

The advantages of the univariate type include the following (Ladha and Deepa, 2011):

- Very fast to compute
- Easily scalable and
- Independency amongst the classifiers.

The disadvantages of the univariate filter include the following (Ladha and Deepa, 2011):

- Ignores interaction with the classifier and
- Ignoring feature dependencies.

The examples of univariate filters include the t-test, euclidian distance and information gain. The examples of multivariate filters include correlation based feature selection (CFS), Markov blanket filter (MBF), fast correlation-based feature selection (FCBF) (Ladha and Deepa, 2011).

3.10.2 The wrapper method

The wrapper method is split into deterministic and randomised type. The examples of deterministic wrapper include the following, sequential forward selection (SFS), sequential backward elimination (SBE), beam search, and Plus L Minus R.

The advantages of the deterministic type include the following (Ladha and Deepa, 2011)**:**

- Easy to implement
- Ability to model feature dependencies and
- Ability to interact with the classifiers.

The disadvantages of the deterministic Wrapper include the following (Ladha and Deepa, 2011)**:**

- Risk of over fitting,
- Likely to converge in local optimum and not a global optimum, and
- Classifier dependent selection.

The examples of randomised wrapper method includes the following, simulated annealing, genetic algorithm, randomised hill climbing, and estimation of distribution algorithm.

The advantage of the randomised wrapper method includes the following (Ladha and Deepa, 2011)**:**

- Converges to global optimum
- Ability to model feature dependencies and
- Ability to interact with classifier.

The disadvantages of the randomised wrapper methods include the following (Ladha and Deepa, 2011)**:**

- Computationally intensive
- classifier dependent selection and
- High risk of over fitting.

3.10.3 The embedded method

The last method search is called the embedded method, the feature selection of the embedded method forms part of the learning algorithm, hence the name embedded. The examples of the embedded methods include the following, decision trees and the weighted naives Bayes feature selection using weigh vector of SVM.

The advantages of the embedded method include the following (Ladha and Deepa, 2011)**:**

- Ability to interact with the classifier,
- Better computational complexity than wrapper methods and
- Ability to model feature dependencies.

The common disadvantage for this type of model search includes the following (Ladha and Deepa, 2011)**:**

- Classifier dependent selection.

The advantages of feature selection are as follows (Frances, 2017):

1. Reducing the dimensionality of the feature space,
2. Limit storage requirements and increase algorithm speed
3. It eliminates the useless, irrelevant or noisy data
4. The key benefits for data analysis tasks are speeding up the running time of the learning algorithms and thus improving computational efficiency.
5. Improving the data quality through removing redundancy in the modelling dataset.
6. Feature selection improves the prediction accuracy of the resulting model
7. Feature set reduction, to save resources in the next round of data collection or during utilization. This is achieved iteratively through a looping process until the noise from the data set is minimized
8. Performance improvement, to gain in predictive accuracy
9. Data comprehension, to gain in-depth understanding about the process that extracted and generated the data or simply visualizes the data

The disadvantages of feature selection are as follows (Frances, 2017):

1. Feature selection does not guarantee improved performance
2. feature that is not useful by itself can be very useful when combined with others
3. if not implemented correctly, there is a possibility to eliminate relevant and useful data
4. feature selection step increase computational time of the model.

4. Methodology and data

A sample of non-financial companies, listed on the JSE was extracted for the period 1997-2017. The dataset was used to build and compare the different prediction models. The logit logistic regression model was built in MS Excel and used as the benchmark. The machine learning models were built using computational algorithms in Python with the scikit-learn library for machine learning. Different variants of each machine learning model were tested to determine the most effective prediction for each variant.

4.1 Input data and data collection

The financial data was sourced from IRESS, IRESS is a technology company providing software and financial data to the financial service industry. The data was extracted in the form of excel csv files. The data used includes both balance sheet and income statement information of all non-financial companies listed in the JSE from the year 1999 to 2017. According to Candice (2016), it is important to ensure that the predictive models are not built on data which would be available after the occurrence of bankruptcy, otherwise the model will produce biased results. This is a total of 994 companies as at December 2017. The extraction process entails the following:

• Gathering of financial data of companies listed in the Johannesburg stock exchange (JSE) from the IRESS website. The data was extracted in the form of excel csv files

• Data was split into three sets, namely: the training set, test set and actual set

• The data for 2017 is used as the actual data, this data does not form part of the data used to train or validate the model

• Because there was no record of bankruptcy in the 2017 data, some of the bankruptcy data was excluded from the training set so that it can be used as the actual data set.

• From the bankruptcy data set, we had 66 companies ranging from year 1999 to 2011.

• The bankruptcy data set from 1999 to 2005 was combined with the training set; there were a total of 42 bankrupt companies and 324 non-bankrupt companies from 1999 to 2016. The total training set and testing set had 366 companies

• Of the data ranging from 1999 to 2016, 70% of the data (i.e.256 companies) is used to build the model, 30% of the data (i.e. 110 companies) is used to validate the model built (i.e. the training set is the remaining 30% and is used to test the model)

• The actual data set had a total of 266 companies, made of 242 non-bankrupt companies and 24 bankrupt companies.

While there are many empirical studies on the best way to split data, 70/30 split are widely used (Raschka, 2016). Raschka (2016) explains that the rule of thumb is that the more development data one has the effective the model will be. This is therefore also the split percentage used in this study.

4.2 Data cleansing and null factor

A NULL value is a value in a field that appears to be blank or no value. If more NULL values is present in dataset as compared to actual number of records, then those records can be removed, otherwise the mean of the field is calculated and the nulls are replaced with the mean values.

4.3 Data sampling and Assumptions.

The work done by Candice (2016) on bankruptcy prediction was leveraged for the benefit of data sampling in this study. Candice (2016) used recursive partitioning to build a financial distress prediction model for JSE companies. Candice (2016) included an indicator for companies which were forced to delist in the JSE during the period 1997-2014 due to financial performance leading to liquidation. The financial information of all bankrupt companies is only available until 12 months before they were delisted from the JSE as a result of financial distress therefore an assumption that a bankrupt company will not have financial information listed with the JSE 12 months after the bankruptcy event has taken place.

IRESS was contacted to help identify the year which the company financial distress occurred. IRESS confirmed that once a company is delisted its financial statements stops to be published. This means that the latest financial statement available for a bankrupt company indicate the year of bankruptcy occurrence. This was important because the financial information 12 months before the occurrence will be used to predict bankruptcy in the next 12 months.

A sample of non-bankrupt companies was extracted for the period 1999-2017. As it is common with many other studies, the problem of missing data was present for this study. Candice (2016)

suggested that one can neglect the missing data or adjust for it on a case by case basis. The problem of missing data was addressed by imputation since deleting the record may impact the parameter estimate. This technique takes the average of the observed values for the variable with data gaps

After cleaning the records with missing data and nulls, the resulting training and test sample of non-bankrupt companies came to a total of 324, while the actual (out of sample) came to 242. The total number of bankrupt companies came to 66. The combined dataset of bankrupt and non-bankrupt companies for the training and test set came to a total of 366 companies, and 266 for the out of sample test.

The financial variables used came to a total of 71 variables, ranging from four categories including profitability, leverage, liquidity and activity. Given that companies can enter into financial distress due to economic slowdown, the period under which the global economy suffered a recession was excluded in this study in order to avoid model bias. This includes the period of 1997-1998 when there was an Asian crises and the period of 2007-2008 when there was a global financial crisis.

4.4 Scope of data

The study only covers South African companies which are listed in the Johannesburg stock exchange (JSE). The investigation excludes financial companies like banks and insurance because they have a different accounting reporting standard to non-financial companies and might contaminate the results. The list of bankrupt and non-bankrupt companies used in the study is available in Appendix A.

4.5 Research strategy applied in building the models

Four different forecasting models were built and compared to determine the best model to use to predict corporate failure for JSE listed non-financial companies. The first model developed was a Logit logistic regression model. This model was used as the benchmark model for prediction accuracy. The predicted benchmark outcomes were then compared to variants of the following three machine learning models:

1. ANN (Artificial neural network);
2. Random forests; and

3. SVM (Support vector machines).

For the machine learning models, supervised learning was chosen, because the desired outcomes are given for the sample dataset and the models were trained to create classifier models by example.

4.6 Statistical method used to build the benchmark model

The logistic regression was built first in in excel. The first task was to generate the parameters for the regression to be used in the final prediction model. The parameters were initially set at 0.1 across all factors. There were 15 variables used in the logistic regression, this resulted in a total of 16 parameters, one intercept coefficient and 15 coefficients. These parameters are used to calculate the logit values which were optimised through the Maximum Log-Likelihood. The solver function in excel was used to perform the optimisation of the parameters used in the model.

4.7 Computational methods used to build machine learning models

Python is one of the most common programming languages for data science and therefore enjoys a large number of useful add-on libraries developed by its great community in both academia and industry (Raschka, 2016). Python language was built for its readability and less complexity (Muller & Guido, 2017). For machine learning tasks, this study makes use of the available scikit-learn library. This library is one of the most common and accessible open source machine learning libraries (Raschka, 2016).

5. Results and analysis

The four models are now discussed in further detail and the findings of each model are outlined. The starting point discusses the benchmark model (i.e. logistic regression) and its results. The second part discuss the findings of the computational machine learning models and also determines whether there are any significant differences in terms of predictive accuracy amongst the three machine learning models. The last part of this chapter compares the three machine learning models against the benchmark model in order to determine the overall effective model for bankruptcy prediction for JSE listed firms.

5.1 Logistic regression results

The first task to fit a logistic regression was to identify the most predictive variable in order to avoid the problem of overfitting. Babyak (2004) explained that overfitted models will fail to reproduce future predictions. Overfitting usually have challenges in predicting future paths of scientific discoveries and findings. Babyak (2004) emphasized the importance and benefit of exploring appropriate candidate predictors in order to overcome the problem of overfitting and spurious results.

Before creating a regression model, the significant financial indicators (variables) had to be identified. This was determined through the mean decrease impurity (MDI) index discussed in section 3.2. The MDI index is used in the random forest technique. The 72 financial variables from the original dataset were ranked from highest to lowest significance and only the top fifteen highest scoring indicators were considered. These variables are listed in table 2 and were included in the final logit logistic regression model. The model yielded a performance of 80% with 74 misclassified companies on the data set of 366 observations.

Table 2: Top 15 variables used in the logistic regression

Variable	**MDI**
Enterprise Value	0.101352
Book Value / Share (c)	0.048778
Cash Flow (Ncta) To All Investments	0.037519
Price / Share (c)	0.037409
Cash Flow / Share (c)	0.036042
NAV / Share (c)	0.030835
Inflation Adjusted Profit / Share (c)	0.030655
Enterprise Value / Share	0.029657
Price / Earnings	0.023332
Retention Rate %	0.020289
Earnings / Share (c)	0.019535
Enterprise Value / PAT	0.018829
Total Assets Turnover	0.017555
Debt / Assets	0.017524

With logistic regression the mean of the response variable Y in terms of an explanatory variable x is modeled relating Y and x through the following equation:

$$Y=\sum_{i=1}^{m} \alpha + b_i x_i, \quad (38)$$

The model results including the parameters from the logistic regression are as follows:

Table 3: Prediction results of the logistic regression

Accuracy rate	**Misclassified Sample**
80%	74

Parameter	**Weight**
α	-0.00084
b1	0.002842
b2	-0.00919
b3	-0.00085
b4	-0.00203
b5	-0.00607
b6	-0.00984
b7	0.002806
b8	-0.00117
b9	0.00039
b10	0.022716
b11	-0.00069
b12	0.013793
b13	-5.30E05
b14	0.000328
b15	0.000356

where the accuracy ratio equals to 1 minus (Number of misclassified samples / total number of observations) *100

The logistic regression was first built using the entire set of variables, the model was not significant and the model performance was zero. Variable importance was introduced through the mean decrease impurity index after realizing that problem and the model performance was zero.

5.2 Machine learning results and analysis

The chapter investigates and analyse various machine learning based models in terms of predictive power which is measured in the form of the model ability to separate good companies from bad companies through an "accuracy ratio" measure. The accuracy ratio is measure of binary misclassification. This is expressed in the form of a ratio. The ratio is simply the number of misclassified pairs to the total number of all pairs in the sample dataset.

5.2.1 ANN results and analysis in python

The final results of the artificial neural network algorithm were achieved by making sure that an appropriate classification algorithm is built. In both industry and academia, it is always recommended that one compares the performance of at least 3 different computational algorithms in order to select the best model for a particular problem because there is no single model that can solve all problems uniformly (Raschka, 2016).

As results, the study shows how the performance of the model has been enhanced by changing from the default parameters of the model function. This process is called fine parameter tuning. According to Raschka (2016), the five main steps that are involved in training a machine learning algorithm can be summarised as follows:

1. Selection of features

2. Choosing a performance metric

3. Choosing a classifier and optimisation algorithm

4. Evaluating the performance of the model and

5. Tuning the algorithm.

The Multi-layer-perceptron classification function (i.e. MLP Classifier) is used to achieve the results. Raschka (2016) states that the default activation function of MLP in sci-kit is the ReLU function which its benefit and advantages has already been discussed in detail in section 3. The MLP is a typical example of a feed forward artificial neural network. The term feed forward refers to the fact that each layer serves as the input to the next layer (Raschka, 2016).

The process followed for this study is inspired by Raschka (2016) and it entails the following three steps:

1. Starting the input layer through forward propagation of the input data in order to generate the output,

2. Calculation of the error from the generated output in order to minimize the cost function, and

3. Back propagation of the error and find its derivative with respect to each weight in the network, and update the model.

5.2.1.1 The process of MLP modelling

The study trained a MLP with seven hyper-parameters, including learning rate, regularization term (i.e. penalties), and activation function, size of hidden layer, maximum number of iterations, maximum number and the solver. Refer to Appendix B for the detail definitions of the parameters involved in the MLP classifier

Modelling results are optimized using a grid search over the input space (Bergstra *et al*, 2011). To ensure that the model converges, the MLP algorithm uses a parameter called the "maximum number of iterations" with an appropriate solver. The solver iterates until convergence based either on a specified tolerance level or the number of iterations (Pedregosa *et al*, 2011). Poor choices of the maximum number of iteration or tolerance level can yield to the error message which states that maximum iteration of 200 reached and the optimisation hasn't converged yet.

The study looks at three different optimisation algorithm of ANN, i.e. stochastic gradient descent (SGD), Adam, and local Broyden Fletcher Goldfarb Shanno (LBFGS).

5.2.1.2 Stochastic gradient descent (SGD)

This technique is frequently used in current artificial learning methods such as convolutional learning and automatic encoders (Yazan and Talu, 2017). Stochastic gradient-based optimisation is the most common optimisation algorithm in many fields of science and engineering (Kingma and Ba, 2014). Problems in these fields can be cast as the optimisation of some scalar parameterised objective function requiring maximization or minimization with respect to its parameters (Kingma and Ba, 2014).

In practice, the learning rate is slowly reduced linearly until the last iteration. If the given objective function is differentiable with respect to its parameters, gradient descent is a relatively a powerful optimisation method since the computation of first-order partial derivatives with respect to all the parameters is of the same computational complexity as just evaluating the function (Kingma and Ba, 2014).

5.2.1.3 Adam method

Adam method of optimisation is for efficient stochastic optimisation that only requires first-order gradients with little memory requirement (Kingma and Ba, 2014). The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. The name Adam is derived from adaptive moment estimation (Kingma and Ba, 2014).

Kingma and Ba (2014) continues to argue that some of the notable powerful traits of the adaptive moment estimates algorithm are that the size of parameter updates remains constant during the rescaling of the gradient, their step sizes are approximately limited by the step size hyper-parameter and they do not require a stationary objective. Adam works with sparse gradients, and it naturally performs a form of step size annealing (Kingma and Ba, 2014). The fundamental idea of Adam is adjusting the learning rate depending on a progress of learning.

5.2.1.4 Limited memory BFGS method

Limited memory BFGS method (LBFGS) is a less computationally intensive method used when the training sample is sufficiently large. The standard L-BFGS method relies on gradient approximations that are not dominated by noise, so that search directions are descent directions, the line search is reliable, and quasi-Newton updating yields useful quadratic models of the objective function (Bollapragada *et al*, 2018).

The LBFGS method is commonly regarded as a batch method in the machine learning community. This is because quasi-Newton algorithms need gradients of high quality in order to construct useful quadratic models and perform reliable line searches for optimisation purposes (Bollapragada *et al*, 2018).

Table 4: MLP results under different hyper-parameter choices for training and test data (1, 1, 1) using a ReLU activation function

Optimiser function	**No of layers**	**Number of Neurons**	**Split Training**	**Total Count**	**Accuracy%**	**Misclassified samples**
Adam	3	(1, 1,1)	70%-30%	366	17%	91
Lbfgs	3	(1, 1,1)	70%-30%	366	No convergence reached	N/A
Sgd	3	(1, 1,1)	70%-30%	366	83%	19

Table 5: MLP results for actual data (1, 1, 1) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(1, 1,1)	266	9%	242
Lbfgs	3	(1, 1,1)	266	No convergence reached	N/A
Sgd	3	(1, 1,1)	266	91%	24

Table 6: MLP results for training and test data (10, 10) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	70%-30%	366	83%	19
Lbfgs	2	(10,10)	70%-30%	366	83%	19
Sgd	2	(10,10)	70%-30%	366	91%	17

Table 7: MLP results for actual data (10, 10) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	266	91%	24
Lbfgs	2	(10,10)	266	91%	24
Sgd	2	(10,10)	266	9%	242

Table 8: MLP results for training and test data (10, 10, 10) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	70%-30%	366	85%	17
Lbfgs	3	(10, 10,10)	70%-30%	366	75%	28
Sgd	3	(10, 10,10)	70%-30%	366	83%	19

Table 9: MLP results for actual data (10, 10,10) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	266	90%	26
Lbfgs	3	(10, 10,10)	266	81%	50
Sgd	3	(10, 10,10)	266	91%	24

Table 10: MLP results for training and test data (10, 10, 10, 10) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	70%-30%	366	79%	23
Lbfgs	4	(10,10,10,10)	70%-30%	366	83%	19
Sgd	4	(10,10,10,10)	70%-30%	366	83%	19

Table 11: MLP results for actual data (10, 10, 10, and 10) using a ReLU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	266	81%	50
Lbfgs	4	(10,10,10,10)	266	91%	24
Sgd	4	(10,10,10,10)	266	91%	24

From the above tables (i.e. table 4-11) obtained using the ReLU activation function, the first two tables represent the training and test results, and the actual data respectively. Table 4 and 5 are the results of the MLP model with an architectural design of three layers with one neuron at each layer. Table 6 and 7 are the results of the MLP model with an architectural design of two layers with ten neurons at each layer. Table 8 and 9 are the results of the MLP model with an architectural design of three layers with ten neurons at each layer. Table 10 and 11 are the results of the MLP model with an architectural design of four layers with ten neurons at each layer.

It is evident from the above tables (i.e. table 4-11) that the architectural design of the neural network plays an important role in the predictive accuracy of the MLP model. The model with one neuron at each layer results in no-convergence problems and low accuracy rate. Whereas the more the number of layers and neuron the better the accuracy. This is because the more the numbers of neurons and lawyers in the model, the different algorithms have more capacity to compute and complete the optimisation processes.

The choice of Adam and LBFGS optimiser functions yield lower accuracy ration and non-convergences, whereas the SGD optimiser function produced the highest results accuracy at 83% when the architectural design is made of three layers with one neuron each. Despite the ease of implementation of the SGD optimiser function, it remains the predominant methodology in training deep learning advocates the use of stochastic gradient descent methods (SGDs) (Le *et al*, 2011).

When the architectural design is made of three layers is made of two layers with ten neurons at each layer (i.e. table 5 and 6). The MLP model with the SGD optimizer I outperformed by the Adam and LBFGS optimiser functions. The SGD has low accuracy ratio of 9% whereas the Adam and the LBFGS both have the accuracy ratio of 91%. However, the SGD still prove its convergence ability despite its prediction ratios.

In the last 4 tables (i.e. table 8-11) with three layers of ten neurons each and four layers of ten neurons each, the MLP with the SGD optimiser function showed high accuracy in the actual data compared to the Adam and the LBFGS. The SGD optimiser function proved itself as an efficient and effective optimisation method that was central in many machine learning success stories, this is consistence with the findings by (Deng et *al*., 2013; Krizhevsky et al., 2012; Hinton & Salakhutdinov, 2006; Hinton *et al*., 2012; Graves *et al*., 2013)

Though they seem to perform better, SGD methods have many disadvantages. According to Le *et al* (2011), the main disadvantage of SGDs is that they require much manual tuning of optimisation parameters such as learning rates and convergence criteria. They are also challenging to develop and debug. An advantage of SGDs is that they are fast for problems that have many training examples and always converge faster.

Table 12: MLP results under different hyper-parameter choices for training and test data (1, 1, 1) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	3	(1, 1,1)	70%-30%	366	84%	18
Lbfgs	3	(1, 1,1)	70%-30%	366	84%	18
Sgd	3	(1, 1,1)	70%-30%	366	83%	19

Table 13: MLP results for actual data (1, 1, 1) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(1, 1,1)	266	91%	23
Lbfgs	3	(1, 1,1)	266	91%	23
Sgd	3	(1, 1,1)	266	91%	24

Table 14: MLP results for training and test data (10, 10) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	70%-30%	366	84%	18
Lbfgs	2	(10,10)	70%-30%	366	66%	37
Sgd	2	(10,10)	70%-30%	366	83%	19

Table 15: MLP results for actual data (10, 10) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	266	90%	26
Lbfgs	2	(10,10)	266	59%	109
Sgd	2	(10,10)	266	91%	24

Table 16: MLP results for training and test data (10, 10, 10) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	70%-30%	366	84%	18
Lbfgs	3	(10, 10,10)	70%-30%	366	66%	37
Sgd	3	(10, 10,10)	70%-30%	366	83%	19

Table 17: MLP results for actual data (10, 10,10) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	266	90%	26
Lbfgs	3	(10, 10,10)	266	59%	109
Sgd	3	(10, 10,10)	266	91%	24

Table 18: MLP results for training and test data (10, 10, 10, 10) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	70%-30%	366	79%	23
Lbfgs	4	(10,10,10,10)	70%-30%	366	83%	19
Sgd	4	(10,10,10,10)	70%-30%	366	83%	19

Table 19: MLP results for actual data (10, 10, 10, and 10) using an ELU activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	266	90%	26
Lbfgs	4	(10,10,10,10)	266	59%	109
Sgd	4	(10,10,10,10)	266	91%	24

From the above tables (i.e. table 12-19) obtained using the ELU activation function, the first two tables represent the training-test results, and the actual data respectively. Table 12 and 13

are the results of the MLP model with an architectural design of three layers with one neuron at each layer. Table 14 and 15 are the results of the MLP model with an architectural design of two layers with ten neurons at each layer. Table 16 and 17 are the results of the MLP model with an architectural design of three layers with ten neurons at each layer. Table 18 and 19 are the results of the MLP model with an architectural design of four layers with ten neurons at each layer.

The choice of the ELU activation function seems to perform poor than the ReLU activation function especially when the Lbfgs optimiser function is used. This can be seen under an architectural design of the three layers of ten neurons each and four layers of ten neurons each (i.e. table 16-19). In aggregate, the ReLU activation function shows better performance under difference optimiser functions to the ELU activation function. The SGD optimiser function has again proved itself as an efficient and effective optimisation method even when combined with the ELU activation function.

Table 20:MLP results under different hyper-parameter choices for training and test data (1, 1, 1) using a softmax activation function

Optimiser function	**No of layers**	**Number of Neurons**	**Split Training**	**Total Count**	**Accuracy%**	**Misclassified samples**
Adam	3	(1, 1,1)	70%-30%	366	83%	19
Lbfgs	3	(1, 1,1)	70%-30%	366	83%	19
Sgd	3	(1, 1,1)	70%-30%	366	83%	19

Table 21: MLP results for actual data (1, 1, 1) using a softmax activation function

Optimiser function	**No of layers**	**Number of Neurons**	**Total Count**	**Accuracy%**	**Misclassified samples**
Adam	3	(1, 1,1)	266	91%	24
Lbfgs	3	(1, 1,1)	266	91%	24
Sgd	3	(1, 1,1)	266	91%	24

Table 22: MLP results for training and test data (10, 10) using a softmax activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	70%-30%	366	83%	19
Lbfgs	2	(10,10)	70%-30%	366	81%	21
Sgd	2	(10,10)	70%-30%	366	83%	19

Table 23: MLP results for actual data (10, 10) using softmax activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	266	91%	24
Lbfgs	2	(10,10)	266	92%	22
Sgd	2	(10,10)	266	91%	24

Table 24: MLP results for training and test data (10, 10, 10) using a softmax activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	70%-30%	366	83%	19
Lbfgs	3	(10, 10,10)	70%-30%	366	81%	21
Sgd	3	(10, 10,10)	70%-30%	366	83%	19

Table 25: MLP results for actual data (10, 10,10) using a softmax activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	266	91%	24
Lbfgs	3	(10, 10,10)	266	92%	22
Sgd	3	(10, 10,10)	266	91%	24

Table 26: MLP results for training and test data (10, 10, 10, 10) using a softmax activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	70%-30%	366	83%	19
Lbfgs	4	(10,10,10,10)	70%-30%	366	84%	18
Sgd	4	(10,10,10,10)	70%-30%	366	83%	19

Table 27: MLP results for actual data (10, 10, 10, and 10) using a softmax activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	266	91%	24
Lbfgs	4	(10,10,10,10)	266	92%	22
Sgd	4	(10,10,10,10)	266	91%	24

From the above tables (i.e. from table 20-27) obtained using the softmax activation function. Different architectural design of the MLP function seems to provide better results than the one obtained under both the ReLU and the ELU activation function. The model performance under training and test data ranges from 81% to 84%, while the model performance under actual ranges from 91% to 92%.

The choice of the softmax activation function seems to perform better than both the ReLU and the ELU activation function especially when the Lbfgs optimiser function is used. The SGD optimiser function has again showed equal performance as the Adam optimiser function under the Softmax activation function. The Lbfgs optimiser function has proven to be the most performing optimiser function under the Softmax activation function. The softmax activation function works better for classification exercises given its logistic nature and its ability to solve binary problems (Nwankpa *et al*, 2018)

Table 28:MLP results under different hyper-parameter choices for training and test data (1, 1, 1) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	3	(1, 1,1)	70%-30%	366	79%	23
Lbfgs	3	(1, 1,1)	70%-30%	366	84%	18
Sgd	3	(1, 1,1)	70%-30%	366	84%	18

Table 29: MLP results for actual data (1, 1, 1) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(1, 1,1)	266	91%	24
Lbfgs	3	(1, 1,1)	266	91%	24
Sgd	3	(1, 1,1)	266	91%	23

Table 30: MLP results for training and test data (10, 10) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	70%-30%	366	84%	19
Lbfgs	2	(10,10)	70%-30%	366	82%	20
Sgd	2	(10,10)	70%-30%	366	84%	24

Table 31: MLP results for actual data (10, 10) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	2	(10,10)	266	91%	25
Lbfgs	2	(10,10)	266	89%	28
Sgd	2	(10,10)	266	90%	26

Table 32: MLP results for training and test data (10, 10, 10) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	70%-30%	366	84%	18
Lbfgs	3	(10, 10,10)	70%-30%	366	82%	20
Sgd	3	(10, 10,10)	70%-30%	366	82%	20

Table 33: MLP results for actual data (10, 10,10) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	3	(10, 10,10)	266	91%	24
Lbfgs	3	(10, 10,10)	266	90%	26
Sgd	3	(10, 10,10)	266	90%	26

Table 34: MLP results for training and test data (10, 10, 10, 10) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Split Training	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	70%-30%	366	83%	19
Lbfgs	4	(10,10,10,10)	70%-30%	366	82%	20
Sgd	4	(10,10,10,10)	70%-30%	366	83%	19

Table 35: MLP results for actual data (10, 10, 10, and 10) using a tanh activation function

Optimiser function	No of layers	Number of Neurons	Total Count	Accuracy%	Misclassified samples
Adam	4	(10,10,10,10)	266	91%	24
Lbfgs	4	(10,10,10,10)	266	89%	28
Sgd	4	(10,10,10,10)	266	91%	24

From the above tables (i.e. table 28-35) obtained using the tanh activation function. Different architectural design of the MLP function seems to provide better results than the one obtained under the ReLU and the ELU activation function. However, the softmax activation functions seems to have better results than the tanh activation function.

The model performance under training and test data for the tanh function from ranges from 82% to 83%, while the model performance under for training and test under the tanh activation function ranges from 83% to 84%.The actual model performance under the tanh activation function ranges from 89% to 91% while the softmax activation function ranges from 91% to 92%.The choice of the softmax activation function seems to perform better than all the three activation functions i.e. (ELU, ReLU and tanh).

5.3 Random forest modelling results

The results for decision trees were achieved using the random forest classifier function in python. A random forest classifier is an estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy while at the same time controls over-fitting (Bergstra *et al*, 2011).

The study trained a decision tree random forest classifier with six hyper-parameters including the following (Bergstra *et al*, 2011).:

- The number of trees in the forest
- Data set split
- Gini function to measure the quality of a split (i.e. Gini impurity and "entropy" for the information gain)
- The maximum depth of the tree
- Minimum number of samples required to split an internal node and
- The minimum number of samples required to be at a leaf node.

Table 36: Classification results of random forest for different data splits

Split Training	**Split Test**	**Total Count**	**Accuracy%**	**Misclassified samples**
70%	30%	366	89%	13
50%	50%	366	86%	26
20%	80%	366	84%	46

Table 37: Classification results of random forest for actual data

Training	**Actual**	**Total Count**	**Accuracy%**	**Misclassified samples**
70%	100%	266	88%	32
50%	100%	266	87%	35
20%	100%	266	84%	36

The results in table 36 and 37 proves that the model produces more accurate results when it is trained on a bigger data set (i.e. 70% training). It is also evident that the performance of the model in terms of prediction accuracy decreases with a decrease in the training set. This behavior is expected in machine learning, because more data is used for training while avoiding overfitting.

For the training and test sample, the highest accuracy comes from splitting the dataset by a 70% for training and 30% for testing. This split yields an accuracy rate of 89% and 13 misclassified samples. Splitting the training set by 50% still performs better compared to a split of 20% with an accuracy rate of 86% and 26 misclassified samples. A 20% split on training data has the lowest accuracy rate of 84% and 46 misclassified samples.

For the actual sample, the highest accuracy also comes from splitting the model built with the training dataset of a 70% for development and 30% for testing. This split yields an accuracy rate of 88% and 32 misclassified samples. This is one percent less than the accuracy score of the test sample (i.e. the out of sample). Testing the actual data on the model trained by splitting the training and test sample into half (i.e. 50%) still gives the second best performance with an accuracy rate of 87% and 35 misclassified samples. A 20% split on training data has the lowest accuracy rate of 84% and 36 misclassified samples.

The results for the best combination of the six hyper-parameters for three of the percentage training vs. testing split are listed in table 37. The results confirmed that the suggested 70/30 split from Raschka *et al* (2016) is more effective, because more data is used for training while avoiding overfitting.

5.4 SVM modelling results

The results for SVM were achieved using the support vector classifier function in python. The main objective of the SVM is to maximize the margin (Raschka, 2016). The margin is defined as the distance between the decision boundary and the training samples that are closest to this hyperplane, which are the so-called support vectors.

The study trained a support vector machine classifier with three hyper-parameters, including the Penalty parameter C of the error term, kernel type to be used in the algorithm, and the tolerance for stopping criterion (Bergstra *et al*, 2011).

Table 38: Classification results of SVM for different kernel types

Kernel	**Split Training**	**Total Count**	**Accuracy%**	**Misclassified samples**
RBF	70%-30%	366	87%	14
linear	70%-30%	366	84%	18
Poly	70%-30%	366	83%	20
Sigmoid	70%-30%	366	82%	20

It is evident from table 38 that the SVC model which uses the RBF kernel yields the best performing outcome in terms of prediction power with an accuracy rate of 87% and 14 misclassified samples. This is followed by the one which uses the linear kernel with an accuracy rate of 84% and 18 misclassified samples. The lowest performing kernel based models are the polynomial and the sigmoid types at 83% and 82% respectively.

Table 39: Classification results of SVM for different kernel types (actual data)

Kernel	**Total Count**	**Accuracy%**	**Misclassified samples**
RBF	266	91%	24
linear	266	91%	24
Poly	266	91%	24
Sigmoid	266	91%	24

Table 39 shows the results of the model when tested on the actual data which it has not seen before (i.e. data was not part of the training or validation set), the model seems to do better

than the data used for the testing set. All kernels types based SVC models seem to perform equally with an accuracy ratio of 91% and a misclassification of 24 samples.

Table 40: Classification results of SVM for different kernel types

Kernel	**Split Training**	**Total Count**	**Accuracy%**	**Misclassified samples**
RBF	50%-50%	183	87%	23
linear	50%-50%	183	89%	21
Poly	50%-50%	183	88%	22
Sigmoid	50%-50%	183	89%	21

When the data for training is spilt into half as shown in table 40, it is clear from the above table 40 that the SVC model which uses the linear and sigmoid kernel yields the best performing outcome in terms of prediction power with an accuracy rate of 89% and 21 misclassified samples. This is followed by the one which uses the polynomial kernel with an accuracy rate of 88% and 22 misclassified samples. The lowest performing kernel based model is the RBF and the at 87% respectively.

Table 41: Classification results of SVM for different kernel types (actual data)

Kernel	**Total Count**	**Accuracy%**	**Misclassified samples**
RBF	266	85%	40
linear	266	85%	40
Poly	266	85%	40
Sigmoid	266	85%	40

When the model which was built on 50% of the training data is tested on the actual data, as shown in table 41, the model drops in performance compared to when the model was built on 70% of the training data. All kernels types based SVC models performs equally with an accuracy ratio of 85% and a misclassification of 40 samples. This is a drop by 6% compared to when the actual is tested on the model built on more data.

5.5 Best performing machine learning model

The performance of the model across all machine learning models in terms of the predictive accuracy ranges from 17% to 92%. On the 70% training-30% test split, the best performing model comes from the ANN with the Softmax activation function. This was observed in an MLP model with an architectural design of four layers with ten neurons at each layer, the model performance under training and test data ranges from 81% to 84%, while the model performance under actual ranges from 91% to 92%.

The random forest classifier is the second highest performing model at 89% and 13 misclassified samples (See table 36), followed by the support vector machine at 87% and 14 misclassified samples (See table 38). The least performing model of all three machine learning models is the ReLU based ANN with an architectural design of three layers and one neuron at each layer. This model has an accuracy rate of 17% for test data and 9% for the actual data, with 91 and 242 misclassified samples respectively (see table 4 and 5).

When considering machine learning techniques, it is clear that machine learning methods can outperform traditional statistical models like the benchmark logistic regression model. Machine learning techniques proved to have the highest accuracy rate compared to statistical models. The ANN model with softmax activation and the Lbfgs optimiser function model yielded the highest score of 92% with 22 misclassified samples (see table 23), compared with the logistic regression which yielded an 80% accuracy rate with 74 misclassified samples (see table 3).

This study proves that another feature that makes machine learning preferred to statistical techniques is their ability to overcome some of the challenges often involved in creating a predictive model such as variable selection, variable importance and missing values (Candice, 2016). The finding of this study also confirms the finding by Louppe *et al* (2013) who also proved that machine learning techniques have the ability to learn and perform better than statistical methods.

6. Conclusion

6.1 Conclusive remarks

The purpose of this study was to investigate and compare the prediction accuracy of both machine learning and traditional methods (i.e. statistical models). The objective was to determine how these models can be exploited in the field of credit risk management to predict both bankruptcy and ultimately credit default in the next 12 months. The study compared the statistical methods and machine learning methods to predict bankruptcy utilising Johannesburg stock exchange listed companies.

The prediction of corporate bankruptcy has been addressed as an increasingly important financial problem and has been extensively analyzed in the accounting literature (Kostopoulos *et al*, 2017). Over recent years, several machine learning methods have been effectively applied to build accurate predictive models for detecting business failure with remarkable results, such as neural networks and ensemble methods (Kostopoulos *et al*, 2017).

When considering machine learning techniques, there is an observation of an additional 12% increase in prediction accuracy when compared to logistic regression (i.e. ANN is at 92% prediction accuracy while logistic regression is 80% prediction accuracy). This is in line with the findings by Wang and Li (2018), Peña, and Martinez (2011), Wang (2017), Barboza (2017) and Bjorn *et al* (2017).

Not all machine learning models performed better than the benchmark logistic regression model, as seen on the ANN model which used the Adam optimiser function. The Adam optimiser based function showed a poor performance of 17% for test data and 9% for the actual data, with 91 and 242 misclassified samples respectively. It is therefore clear that datasets need to be trained correctly with multiple machine learning models and parameters to determine the best model variant to use for each specific case. The ANN model is the most effective and outperformed all models. This finding is consistent with Wagemans (2017), Zibanezhad *et al* (2011), and Lee (2006).

Though the ANN model has some of the least performing model amongst the three machine learning models (i.e. as seen under the ReLU based ANN), the highest accuracy rate still comes from the ANN model after applying the parameter tuning under the Softmax activation model. This proved that a hyper-parameter tuned ANN performs better when compared to the logistic regression model. This finding is consistent with Wang (2017), Kasgari *et al* (2013) and Felix

and Ivan (2015). The conclusion on the superiority of prediction techniques between machine learning methods and statistical methods is that machine learning methods are the most effective when trained correctly and therefore superior to traditional statistical method in bankruptcy prediction.

However, the high performance of machine learning techniques on bankruptcy prediction should not be isolated to an extent where full dependency is placed on machine learning for decision making. The reason for this is because machines can only be smarter to the extent at which they have been given the data to analyse and learn from, some machine learning models may not perform as expected on new set of data or foreign data.

For example, machine learning models trained on Johannesburg stock exchange data may not predict bankruptcy on data from the Zimbabwean stock exchange or any other. This is because of the differences on the determination of bankruptcy, different regulations, listing requirements, etc. As a results, machine learning can only be as effective in decision making if they remain monitored and reviewed by humans who understands the actual problems and how the solutions should be.

6.2 Recommendations for future research

The following areas of research were identified for future work:

- While the current study was based on listed companies, there has been growing interest to understand the credit behaviour of start-up companies falling within the small and medium enterprise (SME) category by both commercial banks and development finance institutions. This study suggests a potential research opportunity on bankruptcy prediction of small and medium enterprises using machine learning.

- Another interesting study will be to investigate how the machine learning techniques: ANN, Decision trees, and SVM can be exploited in the field of finance, specifically in credit risk management to predict bankruptcy for unlisted companies using only qualitative factors. While the current study investigated machine learning techniques for listed firms using quantitative data, there exist a gap to predict bankruptcy for unlisted companies using only qualitative factors.

7. References

Adya, M. and Collopy, F. (1998) How effective are neural networks at forecasting and prediction, a review and evaluation. Journal of Forecasting, 17(1), p481-495.

Al-Osaimy, M. (1998) A neural networks system for predicting Islamic banks performance. JKAU: Econ. and Adm, 11(1), p33-46.

Altman, E.I, and Saunders, A. (1998) Credit risk measurement: Developments over the last 20 years. Journal of Banking and Finance, p1721-1742.

Altman, E.I. (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. Journal of Finance, 4(2), p589-609.

Altman, E.I. (2012) The Altman z-score. INTERNET: http://www.aaii.com/computerizedinvesting/article/the-altman-z-score.pdf. Cited 31 January 2013.

Atiya, F. (2001) Bankruptcy prediction for credit risk using neural networks. IEEE Transactions on Neural Networks, 12(4), p929–935

Babyak, M. (2004) What you see may not be what you get- a brief nontechnical introduction to overfitting in regression-type models. Psychosom Med. 1(66), p411-421.

Baluja, S. (1994) A method for integrating genetic search based function optimisation and competitive learning. School of Computer Science, Carnegie Mellon University, USA.

Barboza F., Kimura H., and Altman E. (2017) Machine learning models and bankruptcy prediction, Expert Systems with Application, 3(1), p405-417.

Beaver, W. (1966) Financial ratios as predictors of failures, empirical research in accounting research. Journal of accounting research, 4(3), p71-111.

Bellovary, D. Giacomino, and M. Akers. (2007) A review of bankruptcy prediction studies 1930-present. Journal of Financial Education, p132, 2007.

Bergstra, J., Bardenet, R., and Bengio, Y. (2011) Algorithms for hyper parameter optimisation. Proceedings of the 24th International Conference on Neural Information Processing Systems, p2546-2554.

Biau, G. (2012) Analysis of a random forests model. The Journal of Machine Learning Research, 13(1), p1063–1095.

Biau, G., Devroye, L., and Lugosi, G. (2008) Consistency of random forests and other averaging classifiers. The Journal of Machine Learning Research, 9(1), p2015–2033.

Bjorn, M., Olof, S., and Andreas D. (2017) Corporate bankruptcy prediction using machine learning, Bachelor's thesis in Economics, University of Gothenburg department of Economics.

Blum, A., and Langley, P. (1997) Selection of relevant features and examples in machine learning. Artificial Intelligence, 97(1-2), p245–271.

Bollapragada, R., Mudigere, D., Nocedal, J., Shi, H-J, and Tang. (2018) A progressive batching L-BFGS method for machine learning.

Breiman, L. (2001) Random forests. Journal Machine learning, 45(1), p5–32.

Breiman, L. (2002) Manual on setting up, using, and understanding random forests v3.1, Statistics Department University of California Berkeley, CA, USA.

Bremer, M. (2012) Multiple linear regression math 261A. INTERNET: http://mezeylab.cb.bscb.cornell.edu/labmembers/documents/supplement%205%20-%20multiple%20regression.pdf. Cited January 2013.

Bruwer, B. W. and Hamman, W. D. (2006) Company failure in South Africa: classification and prediction by means of recursive partitioning. South African journal of business management. 37(4), p7-18.

Candice, S. (2016) The use of recursive partitioning to build a financial distress prediction model for JSE listed equities. Masters financial management, University of Cape Town, South Africa.

Carpenter, G. A., and Grossberg, S. (1988) The art of adaptive pattern recognition by a self-organizing neural network. Computer, 21(3), p77-88.

Chudson, W. (1945) The Pattern of Corporate Financial Structure. New York: National Bureau of Economic Research.

Clevert, D. A., Unterthiner, T. and Hochreiter, S. (2015) Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," arXiv, 2015. [Online]. Available: http://arxiv.org/abs/1511.07289

Cortes, C., and Vapnik, V. (1995) Support-vector network machine learning, Journal of Machine Learning, 3(20), p273–297.

Dahl, G., Sainath, T., and Hinton G. (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout, Department of Computer Science, University of Toronto, New York.

Danenas, P., and Garsva, G. (2015) Selection of support vector machines based classifiers for credit risk domain, Expert Systems with Applications: An International Journal, 42(6), p.3194-3204.

Deakin, A. (1972) A discriminant analysis of predictors of business failure. Journal of Accounting Research, 10 (1), p167-179.

Deng, L., Li, J., Huang, J., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., Gong, Y., and Acero, A. (2013) Recent advances of deep learning for speech research at Microsoft, ICASSP.

Dietterich, T.G. (2000) Ensemble Methods in Machine Learning", *Proc. Conf. Multiple Classifier Systems*, pp. 1-15.

Douglas, R. (2016) A corporate failure prediction model for non-financial South African corporates incorporating best practices used by the credit industry. Masters financial management, University of Cape Town, South Africa.

Du Jardin, P. (2010) Predicting bankruptcy using neural networks and other classification methods: The influence of variable selection techniques on model accuracy, Neurocomputing, 73(10&12), p2047–2060.

Edmister, R, O. (1972) The Journal of Financial and Quantitative Analysis: Supplement: Outlook for the Securities Industry,7(2), pp1477-1493

Efron, B. (1982) The jackknife, the bootstrap and other resampling plans. CBMS-NSF regional conference series in applied mathematics, Society for Industrial and Applied Mathematics (SIAM)

Elloumi, F., and Gueyie′, J. (2001) Financial distress and corporate governance: an empirical analysis. Corporate Governance. The international journal of business in society, 1(1), p15-23.

Fallahpour, S., Lakvan, E.N., Zadeh, M.H. (2017) Using an ensemble classifier based on sequential floating forward selection for financial distress prediction problem. Journal of Retailing and Consumer Services, 34, p159–167

Felix, J., Ivan, P. (2015) Bankruptcy visualization and prediction using neural networks- A study of U.S. commercial banks. Journal of Expert Systems with Applications, 42(6), p2857-2869.

Fisher, R. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, p179-188.

Fitzpatrick, P. (1932) A comparison of ratios of successful industrial enterprises with those of failed companies. The certified public accountant (October, November, December): p598-605, p656-662, p727-731, respectively

Frances, M. (2017) Using Neural Networks and Support Vector Machines for default Prediction in South Africa. Masters of Science, Wits University, South Africa.

Francois, C. (2016) Company financial failure and distress (A perspective). MCom financial management, University Pretoria faculty of Economic and management sciences, South Africa.

Frydman, H., Altman, E. I., and Kao, D. (1985) Introducing recursive partitioning for financial classification: The case of financial distress. The Journal of Finance, 40(1), p26-59.

Gaughan, A. (2015) Mergers, Acquisitions, and Corporate Restructurings. Wiley publishing, fourth Edition. ISBN: 978-0-471-70564-2

Goeletsis Y, Papaloukas C, Exharos T, and Katsis C. (2012) Bankruptcy prediction through artificial intelligence, IGI Global publishers, 3(5), p1-8.

Graves, A., Mohamed, A, R., and, Hinton, G. (2013) Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on pp. 6645–6649. IEEE.

Guyon. (2003) An introduction to variable and feature selection. Journal of machine learning research, 3(1), p1157-1182.

Hastie, T., Tibshirani, R., and Friedman, J. (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer US, first edition.

Haykin, S. (1998) Neural networks: A comprehensive foundation. Prentice Hall, England, 2nd Edition. ISBN-13: 9780132733502

Hinton, G.E. and Salakhutdinov, R, R. (2006) Reducing the dimensionality of data with neural networks. Science, 313 (5786):504–507.

Hofmann, M. (2006) Support vector machines kernels and the kernel trick notes. INTERNET.http://www.cogsys.wiai.unibamberg.de/teaching/ss06/hs_svm/slides/SVM_Seminarbericht_Hofmann.pdf. Cited March 2015.

Hopfield, J, J. (1983) Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci. U.S.*, vol. 79, pp. 2554-2558, Apr. 1982.

Huttunen, H. (2015) Modern machine learning in python, Tampere University of Technology department of signal processing, Finland.

IRESS 2018, IRESS expert website, <http:// http://expert.inetbfa.com/#Logon />.

Jackendoff, N. (1962) A Study of Published Industry Financial and Operating Ratios. Philadelphia: Temple University, Bureau of Economic and Business Research.

Jayanthi, J., Suresh, J., and Vaishnavi, J. (2011) Bankruptcy prediction using SVM and hybrid SVM survey. International Journal of Computer Applications, 3(7), p39 – 45.

Karels, G. and A. Prakash. (1987) Multivariate normality and forecasting of business bankruptcy. *Journal of Business Finance* & *Accounting* 14(4), p573-593.

Karlik, B., and Vehbi, A. (2011) Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks, International Journal of Artificial Intelligence and Expert Systems (IJAE), 1(4), p111–122.

Kartasheva, A, V., and Traskin, M. (2011) Insurers' insolvency prediction using Random Forest classification, Department of Statistics, The Wharton School, University of Pennsylvania.

Kasgari, A., and Salehnezhad, S. (2013) The Bankruptcy Prediction by Neural Networks and Logistic Regression, International Journal of Academic Research in Accounting, Finance and Management Sciences, 3(4), p146-152.

Kim, E. (2013) Everything you wanted to know about the Kernel Trick. INTERNET: http://archive.today/lzSTZ

Kim, J., Weistroer, H. and Redmond, R. (1993) Expert systems for bond rating: a comparative analysis of statistical, rule-based and neural network systems. Expert Systems,10(1), p167-172

Kingma, D., and Ba, J., (2014) Adam: A method for stochastic optimisation. *arXiv preprint arXiv:1412.6980*, 2014.

Kohavi, R. John, G. (1997) Wrappers for feature selection. *Artificial Intelligence*, 97(1-2), p273–324.

Kohavi. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence (IJCAI).

Konirk, S. (2004) Machine learning for corporate failure prediction, an empirical study of south African companies, Degree of Masters in commerce, Department of accounting and department of electrical engineering, University of Cape town, South Africa.

Kostopoulos G., Karlos S., Kotsiantis S., Tampakas V. (2017) Evaluating Active Learning Methods for Bankruptcy Prediction. In: Frasson C., Kostopoulos G. (Eds) Brain Function Assessment in Learning. BFAL 2017. Lecture Notes in Computer Science, vol 10512. Springer, Cham

Kriesel, D. (2005) A brief introduction to neural networks. INTERNET: http://www.dkriesel.com/science/neural_networksWehenkel. Cited June 2007

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012) Imagenet classification with deep convolutional neural networks, INTERNET:http://citeseerx.ist.psu.edu/viewdoc/summary. Cited March 2014.

Kumari, B., and Swarnkar, T. (2011) Filter versus wrapper feature subset selection in large dimensionality micro array: A review. International Journal of Computer Science and Information Technologies, 2(1): p1048-1053

Ladha, L., and Deepa, T. (2011) Feature selection methods and algorithms. International Journal on Computer Science and Engineering (IJCSE), 3(1), p1787-1797

Le, Q., J., Ngiam, A., Coates, A., Lahiri, B., Prochnow., and NG, A. (2011) On Optimisation Methods for Deep Learning, Proc. ICML'11 Proceedings of the 28th International Conference on International Conference on Machine Learning Pages 265-272.

Lee, W. (2006) Genetic programming decision tree for bankruptcy prediction, Proceedings of the 2006 Joint Conference on Information Sciences JCIS 2006, Atlantis Press.

Li, Y.C. and Wang, Y.F. (2018) Machine learning methods of bankruptcy prediction using accounting ratios. Open Journal of Business and Management, 6(1), p1-20.

Louppe, G., Wehenkel, L., and Sutera, A. (2013) Understanding variable importance in forests of randomized trees, University of Liege department of electrical engineering and computer science, Belgium.

Luxton, D., (2016). An Introduction to Artificial Intelligence in Behavioral and Mental Health Care. 10.1016/B978-0-12-420248-1.00001-5.

Maas, A., and Hannun, A. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. Computer Science Department, Stanford University, California, USA.

Marais, M. L., Patel, J., and Wolfson, M. (1984) The experimental design of classification models: An application of recursive partitioning and bootstrapping to commercial bank loan classifications. Journal of Accounting Research, 22(1), p87–113.

Matthew, B. (2017) The Radial Basis Function Kernel, INTERNET: http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/svms/RBFKernel.pdf . Cited January 2018.

McCulloch, W. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5(1), p115-133

Merwin, C. (1942) Financing small corporations in five manufacturing industries, 1926-1936 New York: National Bureau of Economic Research.

Min, J., and Lee, C. (2005) Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Systems with Applications, 28(4), p603–614.

Muller, C., and Guido, S. (2017) Introduction to machine learning with python, O'Reilly media publishing, 1st edition, USA, ISBN-10: 1449369413

Muller, G.H., Steyn-Bruwer, W. S. and Hamman, W. D. (2009) Predicting financial distress of companies listed on the JSE – A comparison of techniques. SA. Journal of Business Management, 40(1), p21-40.

Murat, K., and Kermal, T. (2014) Qualitative bankruptcy prediction rules using artificial Journal of Intelligence techniques, International conference on challenges in IT, engineering and technology, 1(1), p1-5.

Naidu, G. P. and Govinda, K. (2018) Bankruptcy prediction using neural networks, 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, 2018, pp. 248-251.

Nair, V., and Hinton, G. (2010) Rectified linear units improve restricted Boltzmann machines, Haifa, p 807–814. INTERNET:http://dl.acm.org/citation.cfm. Cited August 2018.

Naoum, R, Al-Jaouni, A, and Shaker, M. (2013) A hybrid intrusion detection system using hamming and MAXNET neural nets using NDIS dataset. Journal of emerging trends in computing and information sciences, 4(2), p198-203.

Neal, R. (1992) Connectionist learning of belief networks, Artificial Intelligence. INTERNET:http://doi.org/10.1016/0004-3702 (92)90065-6. Cited August 2017.

Ning, Y., Tianruii, L., and Jing, S. (2007) Construction of decision trees based Entropy and rough sets under tolerance relation. International Conference on Intelligent Systems and KnowledgeEgineering.INTERNE:https://pdfs.semanticscholar.org/bcdd/21f9cabdb0dea8c3a9e7dceb8f1d601536f9.pdf. Cited 7 January 2007.

Nwankpa, C., Ijomah, W., & Gachagan, A. (2018) Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. INTERNET: https://arxiv.org/pdf/1811.03378.pdf. Cited 31 October 2018.

Odom, M. D. and Sharda, R. (1990) A neural network model for bankruptcy prediction, IJCNN International Joint Conference on Neural Networks, San Diego, CA, USA, p163-168 vol.2.

Park, H. (2013) An introduction to logistic regression, Korean Society of Nursing Science publishing, 43(2), p154-164.

Patel, N., and Singh, D. (2015) An algorithm to construct decision tree for machine learning based on similarity factor. International Journal of Computer Applications, 111(10), p22–26.

Pedregosa, F., Varoquaux, G., and Gramfort, A. (2011) Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 1(12), p2825-2830.

Peña, T., and Martinez, S. (2011) Bankruptcy prediction-A Comparison of some statistical and machine learning techniques. INTERNET: http: //www.anterior.banxico.org.mx/publicaciones-y-discursos/publicaciones/documentos-de-investigacion/banxico/%7BB27BAF4A-0741-A858-1DD9-141E38A4F735%7D.pdf. Cited August 2017.

Peng, C., Lee, K., and Ingersoll, G. (2002) An Introduction to logistic regression analysis and reporting. The Journal of Educational Research, 34(7), p1-14.

Pour, E. (2013) Why do companies delist voluntarily from the stock market? Journal of Banking and Finance, p4850-4860.

Radloff, S., and van der Walt, O. (1999) A Combination of a stationary and a nonstationary model for the reduction of Corporate Failure – a New Approach, Unpublished Paper, Rhodes University.

Ramachandran, P., Zoph, B., and Le, Q. (2017) Searching for activation functions. INTERNET:http://arxiv.org/abs/1710.05941. Cited November 2017.

Rao, R, S. (2017) A review on detection and elimination of corporate financial distress. International Journal of Trend in Research and Development, p550-555.

Raschka, S. (2016) Python machine learning, PACKT publishing, Birmingham- Mumbai, 1st Edition. ISBN 978-1-78355-513-0

Refaeilzadeh, P., Tang, L., and Liu, H. (2009) Cross-Validation. Springer US, first edition.

Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(1), p386-408.

Sanaz, S., Farid, S., Ali R.G., and Masoumeh, T. (2016) Bankruptcy prediction using artificial Neural Network. Iranian Journal of Business and Economics, 3(3), p1-16

Sánchez, L., García, V., Marqués, A. (2016) Financial distress prediction using the hybrid associative memory with translation, Journal of Applied Soft Computing, 44(1), p144-152.

Shai, S., and Shai, B. (2014) Understanding machine learning from theory to algorithm, Cambridge University press publishing, New York, 2nd Edition, ISBN 978-1-107-05713-5.

Sharma, H., and Kumar, S. (2016) A Survey on Decision Tree Algorithms of Classification in Data Mining, International Journal of Science and Research, 4(5), p1–4.

Simon, H. (2009) Neural networks and learning Machines. Pearson education publishing, Ontario Canada, 3rd Edition. ISBN-10:0131471392

Sinkey, Jr., J. (1975) A multivariate statistical analysis of the characteristics of problem banks, Journal of finance, 30 (1) p1477-1493

Tsai, C., Hsu, Y., and Yen, D. (2014) A comparative study of classifier ensembles for bankruptcy prediction. Journal of Applied Soft Computing, 24(3), p977-984.

Wagemans F. (2017) Machine learning in bankruptcy prediction, Degree of Masters in commerce, Department of computer science, University of Utrecht.

Wang, N. (2017) Bankruptcy prediction using machine learning. Journal of Mathematical Finance, 7(1), p908-918.

Wehenkel, M., Sutera, A., Bastin, C., and Geurts, P. (2018) Random forests based group importance scores and their statistical Interpretation-Application for Alzheimer's disease. The Journal of the Alzheimer's Association, 14(7), p820-900.

Winakor, A. and Smith, R. (1935) Changes in Financial Structure of Unsuccessful Industrial Corporations. Bureau of Business Research, Bulletin No. 51. Urbana: University of Illinois Press.

Yazan, E., and Talu, M. (2017) Comparison of the stochastic gradient descent based optimisation techniques," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, pp. 1-5. doi: 10.1109/IDAP.2017.8090299

Yeh, C., Chi, D., and Lin, Y. (2014) Going-concern prediction using hybrid random forests and rough set approach, Information Sciences—Informatics and Computer Science, Intelligent Systems, Applications: An International Journal, 254(1), p98-110.

Zeiler, M., Ranzato, M., Monga, R. (2013) on rectified linear units for speech processing, in International Conference on Acoustics, Speech and Signal Processing. IEEE, p3517–3521.

Zhang, X., He, K., and Ren, S. (2015) Delving Deep into Rectifiers surpassing Human-Level performance on ImageNet Classification. INTERNET: http://arxiv.org/abs/1502.01852. Cited August 2016.

Zibanezhad, E., Foroghi, D., and Monadjemi, A. (2011) Applying decision tree to predict bankruptcy. IEEE International Conference on Computer Science and Automation Engineering, p165-169.

Zoubin, G. (2012) Graph-based Semi-Supervised Learning, UK: Department of Engineering University of Cambridge. INTERNET: http://mlg.eng.cam.ac.uk/zoubin/talks/lect3ssl.pdf. Cited November 2016

8. Appendix

Appendix A: List of all companies and variables

Table 42: List of non-bankrupt companies from the JSE between 1997 and 2017

Accelerate Prop Fund Ltd	Attacq Limited	Conduit Capital Ltd	Gold Fields Ltd
Accentuate Limited	Aveng Group Limited	Cons Infrastructure Grp	Grand Parade Inv Ltd
Adaptit Holdings Limited	Avi Ltd	Coronation Fund Mngrs Ld	Grindrod Ltd
Adcock Ingram Hldgs Ltd	Balwin Properties Ltd	Crookes Brothers Ltd	Grit Real Estate Inc Grp
Adcorp Holdings Limited	Barclays Africa Grp Ltd	Csg Holdings Ltd	Group Five Ltd
Adrenna Property Grp Ltd	Barloworld Ltd	Curro Holdings Limited	Growthpoint Prop Ltd
Advanced Health Ltd	Basil Read Holdings Ltd	Datatec Ltd	Hammerson Plc
Advtech Ltd	Bauba Platinum Limited	Delta Emd Ltd	Harmony Gm Co Ltd
Aeci Limited	Bell Equipment Ltd	Delta Property Fund Ltd	Homechoice Int Plc
African & Over Ent Ltd	Bhp Billiton Plc	Deneb Investments Ltd	Hosken Cons Inv Ltd
African Dawn Capital Ltd	Bid Corporation Ltd	Discovery Ltd	Hospitality Prop Fund B
African Equity Emp Inv L	Bidvest Ltd	Distell Group Ltd	Howden Africa Hldgs Ltd
African Media Ent Ltd	Blue Label Telecoms Ltd	Distr And Warehousing	Hudaco Industries Ltd
African Oxygen Limited	Bowler Metcalf Ltd	Drd Gold Ltd	Huge Group Ltd
African Phoenix Inv Ltd	Brait Se	E Media Holdings Ltd	Hulamin Ltd
African Rainbow Min Ltd	Brikor Ltd	Echo Polska Prop N.V.	Hulisani Limited
Afrimat Limited	Brimstone Inv Corp Ltd	Efficient Group Ltd	Hwange Colliery Ltd
Afrocentric Inv Corp Ltd	British American Tob Plc	Efora Energy Limited	Hyprop Inv Ltd
Ah-Vest Limited	Buffalo Coal Corp	Elb Group Ltd	Imbalie Beauty Limited
Alaris Holdings Ltd	Cafca Limited	Ellies Holdings Ltd	Impala Platinum Hlgs Ltd
Alexander Forbes Grp Hld	Calgro M3 Hldgs Ltd	Emira Property Fund Ltd	Imperial Holdings Ltd
Allied Electronics Corp	Capevin Holdings Ltd	Enx Group Limited	Indequity Group Ltd
Alviva Holdings Ltd	Capital & Regional Plc	Eoh Holdings Ltd	Indluplace Properties Lt
Anchor Group Limited	Capital Appreciation Ltd	Equites Prop Fund Ltd	Ingenuity Property Inv
Andulela Inv Hldgs Ltd	Capital&Counties Prop Pl	Erin Energy Corporation	Insimbi Ref & Alloy Sup
Anglo American Plat Ltd	Capitec Bank Hldgs Ltd	Esor Limited	Interwaste Hldgs Ltd
Anglo American Plc	Cargo Carriers Ltd	Extract Group Limited	Intu Properties Plc
Anglogold Ashanti Ltd	Cashbuild Ltd	Exxaro Resources Ltd	Investec Australia Prop
Anheuser-Busch Inbev Sa	Caxton Ctp Publish Print	Fairvest Property Hldgs	Investec Ltd
Ansys Limited	Central Rand Gold Ltd	Famous Brands Ltd	Investec Plc
Arb Holdings Limited	Choppies Enterprises Ltd	Ferrum Crescent Limited	Investec Property Fund L
Arcelormittal Sa Limited	Chrometco Ltd	Finbond Group Ltd	Invicta Holdings Ltd
Argent Industrial Ltd	City Lodge Hotels Ltd	Firstrand Ltd	Ipsa Group Plc
Ascendis Health Ltd	Clicks Group Ltd	Fortress Reit Ltd A	Isa Holdings Limited
Aspen Pharmacare Hldgs L	Clientele Ltd	Gaia Infrastruct Cap Ltd	Italtile Ltd
Assore Ltd	Clover Industries Ltd	Gemgrow Properties Ltd A	Jasco Electron Hldgs Ltd
Astoria Investments Ltd	Cognition Holdings Ltd	Glencore Plc	Jse Ltd
Astral Foods Ltd	Comair Limited	Global Asset Mngment Ltd	Jubilee Metals Group Plc
Atlantic Leaf Prop Ltd	Combined Motor Hldgs Ltd	Globe Trade Centre S.A.	Kap Industrial Hldgs Ltd
Atlatsa Resources Corp	Compagnie Fin Richemont	Gold Brands Inv Ltd	Kaydav Group Ltd

Table 43: List of non-bankrupt companies from the JSE between 1997 and 2017

Kibo Mining Plc	Novus Holdings Limited	Reunert Ltd	Tharisa Plc
Kumba Iron Ore Ltd	Nutritional Holdings Ltd	Rex Trueform Group Ltd	The Foschini Group Limit
Labat Africa Ltd	Nu-World Hldgs Ltd	Rhodes Food Grp Hldg Ltd	The Spar Group Ltd
Lewis Group Ltd	Nvest Financial Hldgs Lt	Rmb Holdings Ltd	Tiger Brands Ltd
Liberty Holdings Ltd	Oando Plc	Rockwell Diamonds Inc	Tiso Blackstar Group Se
Liberty Two Degrees	Oasis Crescent Prop Fund	Rolfes Technology Hldgs	Tongaat Hulett Ltd
Life Healthc Grp Hldgs L	Oceana Group Ltd	Royal Bafokeng Platinum	Torre Industries Limited
London Fin Inv Group Plc	Octodec Invest Ltd	Sa Corp Real Estate Ltd	Tower Property Fund Ltd
Lonmin Plc	Old Mutual Plc	Sabvest Ltd	Tradehold Ltd
Marshall Monteagle Plc	Omnia Holdings Ltd	Safari Investments Rsa L	Trans Hex Group Ltd
Mas Real Estate Inc.	Onelogix Group Ltd	Sanlam Limited	Transaction Capital Ltd
Massmart Holdings Ltd	Orion Real Estate Ltd	Santam Limited	Transcend Res Prop Fd Ld
Master Drilling Grp Ltd	Pallinghurst Res Ltd	Santova Logistics Ltd	Transpaco Ltd
Mazor Group Ltd	Pan African Resource Plc	Sappi Ltd	Trellidor Hldgs Ltd
Mc Mining Limited	Pbt Group Limited	Sasfin Holdings Ltd	Trematon Capital Inv Ltd
Mediclinic Int Plc	Pembury Lifestyle Grp Lt	Sasol Limited	Trencor Ltd
Merafe Resources Ltd	Peregrine Holdings Limit	Schroder Eur Reit Plc	Trustco Group Hldgs Ltd
Metair Investments Ltd	Phumelela Game Leisure	Sephaku Holdings Ltd	Truworths Int Ltd
Metrofile Holdings Ltd	Pick N Pay Stores Ltd	Shoprite Holdings Ltd	Tsogo Sun Holdings Ltd
Micromega Holdings Ltd	Pioneer Foods Group Ltd	Sibanye Gold Limited	Unicorn Capital Pnr Ltd
Mine Restoration Inv Ltd	Ppc Limited	Silverbridge Holdings	Value Group Ltd
Mix Telematics Ltd	Primeserv Group Ltd	Sirius Real Estate Ltd	Verimark Holdings Ltd
Mmi Holdings Limited	Psg Group Ltd	South Ocean Holdings Ltd	Visual International Hld
Mondi Ltd	Psg Konsult Limited	South32 Limited	Vodacom Group Ltd
Mondi Plc	Psv Holdings Ltd	Spanjaard Limited	Vukile Property Fund Ltd
Mpact Limited	Purple Group Ltd	Spur Corporation Ltd	Vunani Ltd
Mr Price Group Ltd	Putprop Ltd	Standard Bank Group Ltd	Wescoal Holdings Ltd
Mtn Group Ltd	Quantum Foods Hldgs Ltd	Stefanuti Stck Hldgs Ltd	Wesizwe Platinum Ltd
Murray & Roberts Hldgs	Rand Merchant Inv Hldgs	Steinhoff Int Hldgs N.V.	Wg Wearne Ltd
Mustek Ltd	Randgold & Expl Co Ltd	Stellar Cap Partners Ltd	Wilderness Holdings Ltd
Nampak Ltd	Raubex Group Ltd	Stenprop Limited	Wilson Bayly Hlm-Ovc Ltd
Naspers Ltd -N-	Rcl Foods Limited	Stor-Age Prop Reit Ltd	Woolworths Holdings Ltd
Nedbank Group Ltd	Rdi Reit P.L.C	Stratcorp Ltd	Workforce Holdings Ltd
Net 1 Ueps Tech Inc	Rebosis Property Fund Lt	Sun International Ltd	York Timber Holdings Ltd
Netcare Limited	Redefine Properties Ltd	Super Group Ltd	Zeder Inv Ltd
New Frontier Prop Ltd	Reinet Investments S.C.A	Taste Hldgs Ltd	
Newpark Reit Ltd	Remgro Ltd	Tawana Resources Nl	
Nictus Ltd	Renergen Limited	Telemaster Holdings Ltd	
Niveus Investments Ltd	Resilient Reit Limited	Telkom Sa Soc Ltd	
Northam Platinum Ltd	Resource Generation Ltd	Texton Property Fund Ltd	

Table 44: List of non-bankrupt companies from the JSE between 1997 and 2017 identified by Candice (2016)

Advanced Tech Syst Ord	Millionair Charter Ltd
Afribrand Holdings Ltd	Net 1 Applied Technology
Aludie Ltd	New Africa Invest Ld Ord
Ambit Properties Ltd	Nimbus Holdings Ltd
Amlac Ltd	Ninian And Lester Hldgs
Aps Technologies Ltd	Omega Alpha Internationl
Aquila Growth Ltd	Pacific Hldgs Ltd
Beget Holdings Ltd	Pals Holding Ltd
Best Cut Ltd	Pasdec Resources Sa Ltd
Bolton Industrial Hldgs	Queensgate Hot & Leis Lt
Bryant Technology Ltd	Retail Apparel Group Ltd
Cci Holdings Ltd	Roadcorp Ltd
Celcom Group Limited	Set Point Group Limited
Century Carbon Mining Ld	Siltek Ltd
Conafex Hldgs Socie Anon	Stocks Hotels And Resort
Core Holdings Ltd	The Laser Group Ltd
Country Foods Ltd	Tiger Wheels Ltd
Dialogue Group Hldgs Ltd	Top Info Technology Hldg
Dna Supply Chain Investm	Unispin Holdings Ltd
Dynamo Retail Ltd	Unitrans Ltd
Ec-Hold Ltd	Universal Growth Hldgs
Elementone Ltd	Venter Leisure And Comm
Enterprise Risk Mngment	Voltex Holdings Ltd Ord
Eureka Ind Ltd Ord	Winecorp Ltd
Fashion Africa Ltd	Zarara Energy Ltd
Frontrange Limited	
Gencor Ltd	
Global Technology Ltd	
Global Village Hldgs Ltd	
Glodina Holdings Ltd	
Harwill Invest Ltd	
Idion Technology Hldgs	
Intertrading Ltd	
Kairos Industrial Hldgs	
Kelgran Ltd	
King Consolidated Hldgs	
Kirchmann-Hurry Prop Ltd	
Masterfridge Ltd	
Mathomo Group Ltd	
Messina Ltd	

Table 45: List of financial variables used in the study

Assets / Capital Employed	Return On Average Equity %
Book Value / Share (c)	Return On Average External Investments %
Cash Flow / Share (c)	Return On Capital Employed %
Accounts Receivable / Turnover	Return On Equity %
Cash Flow Dividend Cover	Return On External Investments %
Cash Flow Interest Cover	Total Assets Turnover
Current Ratio	Total Debt / Cash Flow
Debt / Assets	Turnover / Employee
Debt / Equity	Adequacy Ratio
Directors Remuneration % Profit Before Tax	Cash Flow (Cata Less Pref. Dividend) / Share
Dividend / Share (c)	Cash Flow (Cata) To Current Liabilities
Dividend Cover	Cash Flow (Cata) To Net Earnings After Tax
Dividend Yield %	Cash Flow (Cata) To Total Debt
Earnings / Share (c)	Cash Flow (Cata) To Turnover (Margin)
Earnings Yield %	Cash Flow (Ncta) To All Investments
Inflation Adjusted Profit / Share (c)	Cash Flow (Ncta) To Capital Investments
Inflation Adjusted Return On Assets %	Cash Flow (Ncta) To Financial Investments
Inflation Adjusted Return On Average Equity %	Cash Flow Less Interest Paid To Income Before Tax
Inflation Adjusted Return On Average Total Assets %	Cash Flow Return On Total Net Assets
Inflation Adjusted Return On Equity %	Attributable Earnings
Interest Cover	Dividend / Share
Leverage Factor	Earnings / Share
Long Term Loans % Total Debt	Growth In Cash From Operating Activities
NAV / Share (c)	Net Turnover
Net Profit Margin %	Total Assets
Operating Profit / Employee	Total Shareholders Interest
Operating Profit Margin %	Enterprise Value
Price / Book Value	Enterprise Value / Cash
Price / Cash	Enterprise Value / EBIT
Price / Cash Flow	Enterprise Value / EBITDA
Price / EBIT	Enterprise Value / PAT
Price / EBITDA	Enterprise Value / Share
Price / Earnings	
Price / Inflation Adjusted Profit	
Price / NAV	
Price / Share (c)	
Quick Ratio	
Retention Rate %	
Return On Assets %	
Return On Average Assets %	

Table 46: List of financial variables used in the study

Variable	Definition
Assets / Capital Employed	Fixed Assets/ Capital Employed Ratio indicate the extent to which the long term funds are sunk in fixed assets which are supplied by creditors and owners of the firm.
Book Value / Share (c)	Book value of equity per share indicates a firm's net asset value (total assets - total liabilities) on a per-share basis. When a stock is undervalued, it will have a higher book value per share in relation to its current stock price in the market
Cash Flow / Share (c)	Cash flow per share is the after-tax earnings plus depreciation on a per-share basis that functions as a measure of a firm's financial strength
Accounts Receivable / Turnover	Accounts receivable turnover is the number of times per year that a business collects its average accounts receivable. The ratio is used to evaluate the ability of a company to efficiently issue credit to its customers and collect funds from them in a timely manner
Cash Flow Dividend Cover	The Dividend Coverage Ratio, also known as dividend cover, is a financial metric that measures the number of times that a company can pay dividends to its shareholders. ... While it is arrived at through the income statement, the net profit is also used in both the balance sheet and the cash flow statement
Cash Flow Interest Cover	The cash flow coverage ratio is a liquidity ratio that measures a company's ability to pay off its obligations with its operating cash flows. ... It is reflected as a multiple, illustrating how many times over earnings can cover current obligations like rent, interest on short term notes and preferred dividends
Current Ratio	The current ratio is a liquidity ratio that measures a company's ability to pay short-term obligations or those due within one year. It tells investors and analysts how a company can maximize the current assets on its balance sheet to satisfy its current debt and other payables.
Debt / Assets	Debt Ratio is a financial ratio that indicates the percentage of a company's assets that are provided via debt. It is the ratio of total debt (long-term liabilities) and total assets (the sum of current assets, fixed assets, and other assets such as 'goodwill')
Debt / Equity	The debt-to-equity (D/E) ratio is calculated by dividing a company's total liabilities by its shareholder equity. These numbers are available on the balance sheet of a company's financial statements. The ratio is used to evaluate a company's financial leverage

Directors Remuneration % Profit Before Tax	The ratio of Directors Remuneration as a percentage of Profit Before Tax
Dividend / Share ©	The term dividends per share (DPS) refers to the total dividend a company pays out over a 12-month period, divided by the total number of outstanding shares

Variable	Definition
Dividend Cover	Dividend cover, also commonly known as dividend coverage, is the ratio of company's earnings (net income) over the dividend paid to shareholders, calculated as net profit or loss attributable to ordinary shareholders by total ordinary dividend
Dividend Yield %	A dividend expressed as a percentage of a current share price
Earnings / Share (c)	Earnings Per Share represents the portion of a company's profit allocated to each outstanding share of common stock. The net income (reported or estimated) for a period divided by the total number of shares outstanding (TSO) during that period
Earnings Yield %	The earnings yield refers to the earnings per share for the most recent 12-month period divided by the current market price per share. The earnings yield (which is the inverse of the P/E ratio) shows the percentage of each rand invested in the stock that was earned by the company
Inflation Adjusted Profit / Share ©	The inflation-adjusted return is the measure of return that takes into account the time period's inflation rate allocated to each outstanding share of common stock
Inflation Adjusted Return On Assets %	The inflation-adjusted return on assets is the measure of return that takes into account the time period's inflation rate. Inflation-adjusted return on assets reveals the return on an assets after removing the effects of inflation
Inflation Adjusted Return On Average Equity %	The inflation-adjusted return on average common stock is the measure of return that takes into account the time period's inflation rate. Inflation-adjusted return on average common stock reveals the return on an assets after removing the effects of inflation
Inflation Adjusted Return On Average Total Assets %	The inflation-adjusted return on total assets is the measure of return that takes into account the time period's inflation rate. Inflation-adjusted return on assets reveals the return on

	an total assets after removing the effects of inflation
Inflation Adjusted Return On Equity %	The inflation-adjusted return on common stock is the measure of return that takes into account the time period's inflation rate. Inflation-adjusted return on common stock reveals the return on an assets after removing the effects of inflation
Interest Cover	The interest coverage ratio is used to determine how easily a company can pay their interest expenses on outstanding debt.

Variable	Definition
Leverage Factor	A leverage ratio is any one of several financial measurements that look at how much capital comes in the form of debt (loans) or assesses the ability of a company to meet its financial obligations
Long Term Loans % Total Debt	Long-term debt % Total debt consists of loans and financial obligations lasting over one year. Long-term debt for a company would include any financing or leasing obligations that are to come due after a 12-month period
NAV / Share (c)	Net asset value per share. The value of a single share of a mutual fund. Calculated by subtracting the fund's total LIABILITIES from its total ASSETS and dividing the result by the OUTSTANDING SHARES. Also known as Average Price
Net Profit Margin %	Net profit margin is the percentage of revenue remaining after all operating expenses, interest, taxes and preferred stock dividends (but not common stock dividends) have been deducted from a company's total revenue
Operating Profit / Employee	Revenue per employee (also called sales per employee) is a financial ratio that measures the revenue generated by each employee of the company on average
Operating Profit Margin %	The operating margin measures how much profit a company makes on a rand of sales, after paying for variable costs of production, such as wages and raw materials, but before paying interest or tax. It is calculated by dividing a company's operating profit by its net sales
Price / Book Value	The Price to Book Ratio formula, sometimes referred to as the market to book ratio, is used to compare a company's net assets available to common shareholders relative to the sale price of its stock. The formula for

	price to book value is the stock price per share divided by the book value per share.
Price / Cash	The price-to-cash ratio is a stock valuation indicator or multiple that measures the value of a stock's price relative to its cash per share.
Price / Cash Flow	The price-to-cash flow (P/CF) ratio is a stock valuation indicator or multiple that measures the value of a stock's price relative to its operating cash flow per share. The ratio uses operating cash flow which adds back non-cash expenses such as depreciation and amortization to net income
Price / EBIT	This is the sum of a company's equity value or market capitalization plus its debt less cash. EV is typically used in buyouts. The EV/EBIT ratio is calculated by dividing EV by EBIT to achieve an earnings multiple that is more comprehensive than the P/E ratio

Variable	Definition
Price / EBITDA	This is the sum of a company's equity value or market capitalization plus its debt less cash. EV is typically used in buyouts. The EV/EBITDA ratio is calculated by dividing EV by EBITDA to achieve an earnings multiple that is more comprehensive than the P/E ratio
Price / Earnings	In essence, the price-to-earnings ratio indicates the rand amount an investor can expect to invest in a company in order to receive one rand of that company's earnings. This is why the P/E is sometimes referred to as the price multiple because it shows how much investors are willing to pay per rand of earnings
Price / Inflation Adjusted Profit	Real prices are defined as prices that have been adjusted for inflation, price per inflation adjusted is the price of the share divided by the calculated inflation adjusted profit
Price / NAV	An NAV computation is undertaken once at the end of each trading day based on the closing market prices of the portfolio's securities
Price / Share (c)	The price per share calculated as the total dividends paid per year, divided by the market price of the stock. This is the return on investment to investors if they were to buy the shares at the current market price

Quick Ratio	The quick ratio is an indicator of a company's short-term liquidity position and measures a company's ability to meet its short-term obligations with its most liquid assets
Retention Rate %	Employee retention rate is a helpful statistic for an employer to calculate – both as a benchmark and periodically (ex: quarterly or bi-annually). The formula is simple. Divide the number of employees who left during a period by the total number of employees at the end of a period to get the percentage
Return On Assets %	Return on assets (ROA) is a financial ratio that shows the percentage of profit a company earns in relation to its overall resources. It is commonly defined as net income divided by total assets
Return On Average Assets %	It is also known as simply return on assets (ROA). The ratio shows how well a firm's assets are being used to generate profits. ROAA is calculated by taking net income and dividing it by average total assets. The final ratio is expressed as a percentage of total average assets
Return On Average Equity %	The return on average equity is a financial ratio that measures the profitability of a company in relation to the average shareholders' equity. This financial metric is expressed in the form of a percentage which is equal to net income after tax divided by the average shareholders' equity for a specific period of time

Variable	Definition
Return On Average External Investments %	Similar to Return on Investment (ROI) is a performance measure used to evaluate the efficiency of an investment or compare the efficiency of a number of different investments
Return On Capital Employed %	Return on capital employed formula is calculated by dividing net operating profit or EBIT by the employed capital
Return On Equity %	Return on equity (ROE) is a measure of financial performance calculated by dividing net income by shareholders' equity. Because shareholders' equity is equal to a company's assets minus its debt, ROE could be thought of as the return on net assets
Return On External Investments %	Similar to Return on Investment (ROI) is a performance measure used to evaluate the

	efficiency of an investment or compare the efficiency of a number of different investments
Total Assets Turnover	The asset turnover ratio measures the efficiency of a company's assets to generate revenue or sales. It compares the dollar amount of sales or revenues to its total assets. The asset turnover ratio calculates the net sales as a percentage of its total assets
Total Debt / Cash Flow	Free Cash Flow to Debt is a ratio that shows the fraction of all debt that would be repaid in one year if all of the free cash flow went to repaying debt. It allows investors to see the company's financial stability
Turnover / Employee	Revenue Per Employee Ratio Formula. Revenue per employee is an important financial ratio calculated by dividing revenues generated for a specific period by the number of employees in a company. It helps as a measure of average financial productivity for each employee of the company
Adequacy Ratio	The capital adequacy ratio (CAR) is a measurement of a bank's available capital expressed as a percentage of a bank's risk-weighted credit
Cash Flow (Cata) To Current Liabilities	The operating cash flow ratio is a measure of the number of times a company can pay off current debts with cash generated within the same period. A high number, greater than one, indicates that a company has generated more cash in a period than what is needed to pay off its current liabilities
Cash Flow (Cata) To Net Earnings After Tax	Cash flow after taxes (CFAT) is a measure of financial performance that looks at the company's ability to generate cash flow through its operations. It is calculated by adding back non-cash charges such as amortization, depreciation, restructuring costs and impairment to net income
Cash Flow (Cata) To Turnover (Margin)	Inventory turnover is an efficiency/activity ratio which estimates the number of times per period a business sells and replaces its entire batch of inventories

Variable	Definition

Cash Flow (Ncta) To All Investments	Cash flow from investing activities is an item on the cash flow statement that reports the aggregate change in a company's cash position resulting from investment gains or losses and changes resulting from amounts spent on investments in capital assets, such as plant and equipment
Cash Flow (Ncta) To Capital Investments	Cash flow from investing activities is an item on the cash flow statement that reports the aggregate change in a company's cash position resulting from investment gains or losses and changes resulting from amounts spent on investments in capital assets
Cash Flow (Ncta) To Financial Investments	Cash flow from investing activities is an item on the cash flow statement that reports the aggregate change in a company's cash position resulting from investment gains or losses and changes resulting from amounts spent on financial investments
Cash Flow Less Interest Paid To Income Before Tax	Cash Flow Less Interest paid To income Before Tax
Cash Flow Return On Total Net Assets	The cash return on assets (cash ROA) ratio is used to compare a business's performance with that of other industry members. It is an efficiency ratio that rates actual cash flows to company assets without being affected by income recognition or income measurements
Attributable Earnings	Definition of attributable profit/loss. A term sometimes used to denote a company's bottom line profit or loss, i.e. the full profit or loss that can be attributed to a company and from which shareholders may receive a dividend
Dividend / Share	Dividend per share (DPS) is the sum of declared dividends issued by a company for every ordinary share outstanding. The figure is calculated by dividing the total dividends paid out by a business, including interim dividends, over a period of time by the number of outstanding ordinary shares issued
Earnings / Share	Earnings per share are the portion of a company's profit that is allocated to each outstanding share of its common stock. It is calculated by taking the difference between a company's net income and dividends paid for preferred stock and then dividing that figure by the average number of shares outstanding

Variable	Definition
Growth In Cash From Operating Activities	Increase in cash from operating activities usually refers to the 12-month increase on the first section of the statement of cash flows. Cash from operating activities focuses on the cash inflows and outflows from a company's main business activities of buying and selling merchandise, providing services, etc.
Net Turnover	Net turnover is a value that can express a number of different figures in business. A company may use net turnover to measure the total volume of sales as well as the influx of new employees
Total Assets	Total assets refer to the total amount of assets owned by a person or entity. Assets are items of economic value, which are expended over time to yield a benefit for the owner
Total Shareholders' Interest	Total Shareholders' Interest, also referred to as a stockholder, is any person, company, or institution that owns at least one share of a company's stock. As equity owners, shareholders are subject to capital gains (or losses) and/or dividend payments as residual claimants on a firm's profits
Enterprise Value	Enterprise value (EV) is a measure of a company's total value, often used as a more comprehensive alternative to equity market capitalization. EV includes in its calculation the market capitalization of a company but also short-term and long-term debt as well as any cash on the company's balance sheet
Enterprise Value / Cash	Enterprise Value / Free Cash Flow. Enterprise Value to Free Cash Flow compares the total valuation of the company with its ability to generate Cashflow. It is the inverse of the Free Cash Flow Yield
Enterprise Value / EBIT	The enterprise-value-to-EBITDA ratio is calculated by: EV divided by EBIT or earnings before interest and taxes. EV (the numerator) is the company's enterprise value (EV) and is calculated as follows: EV = market capitalization + preferred shares + minority interest + debt - total cash

Enterprise Value / EBITDA	The enterprise-value-to-EBITDA ratio is calculated by: EV divided by EBITDA or earnings before interest, taxes, depreciation, and amortization. EV (the numerator) is the company's enterprise value (EV) and is calculated as follows: EV = market capitalization + preferred shares + minority interest + debt - total cash
Enterprise Value / PAT	The enterprise-value-to-PAT ratio is calculated by: EV divided by Profit after Tax. EV (the numerator) is the company's enterprise value (EV) and is calculated as follows: EV = market capitalization + preferred shares + minority interest + debt - total cash
Enterprise Value / Share	Enterprise value per share ratio, it is calculated by multiplying the number of equity shares outstanding by the price of the stock. It completely ignores debt capital

Appendix B: Multi-Layer Perceptron (MLP) code

```
#preparation of ranking
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

#Loading of data
raw_data = pd.read_excel(r"Modelling data_4.xlsx")
#raw_data = raw_data.fillna(0)
y = raw_data['Bankrupt']
X = raw_data

# split into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size = 0.3, random_state = 42)

del X['Bankrupt']

#preparation of ranking
from sklearn.feature_selection import SelectFromModel
forest=RandomForestClassifier(n_estimators=1000,random_state=0,n_jobs=-1)
sfm = SelectFromModel(forest, threshold=0.0003)
sfm.fit( X_train , y_train )

# cross validation and parameter tuning
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
import itertools

parameters={
'hidden_layer_sizes':  [x for x in itertools.product((100,150),repeat=2)]
}

# training  the model using the  training dataset
MLPClass = MLPClassifier()
clf= GridSearchCV(estimator=MLPClass,param_grid=parameters)
results = clf.fit(X_train, y_train)

print("Best: %f using %s" % (results.best_score_, results.best_params_))

clf = MLPClassifier(solver='adam', alpha=1e-5,
hidden_layer_sizes=list(results.best_params_['hidden_layer_sizes']), random_state=1, batch_size=50,
max_iter=200)
clf.fit(X_train, y_train)

# predicting the outcome of the test sample using the optimal model
y_pred_test = clf.predict(X_test)
print('Misclassified samples:%d' % (y_test != y_pred_test).sum())
print('Accuracy Neuaral Network:%.2f' % accuracy_score(y_pred_test,y_test))
```

```
# feature importance and ranking power of the predictors
forest.fit(X_train, y_train )
feat_labels=X_train.columns

# Print the name and gini importance of each feature

for feature in zip(feat_labels, forest.feature_importances_):
print(feature)

for feature_list_index in sfm.get_support(indices=True):

print(feat_labels[feature_list_index])
```

Appendix C: Decision tree classifier code

```
#preparation of ranking
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

#Loading of data
raw_data = pd.read_excel(r"Modelling data_4.xlsx")
#raw_data = raw_data.fillna(0)
y = raw_data['Bankrupt']
X = raw_data

# split into train and test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
      X, y, test_size = 0.3, random_state = 42)

del X['Bankrupt']

#preparation of ranking
from sklearn.feature_selection import SelectFromModel
forest=RandomForestClassifier(n_estimators=1000,random_state=0,n_jobs=-1)
sfm = SelectFromModel(forest, threshold=0.0003)
sfm.fit( X_train , y_train )

# cross validation and parameter tuning
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
import itertools

parameters={
'hidden_layer_sizes':  [x for x in itertools.product((100,150),repeat=2)]
}

# training  the model using the  training dataset
MLPClass = MLPClassifier()
clf= GridSearchCV(estimator=MLPClass,param_grid=parameters)
results = clf.fit(X_train, y_train)

print("Best: %f using %s" % (results.best_score_, results.best_params_))

clf = MLPClassifier(solver='adam', alpha=1e-5,
                    hidden_layer_sizes=list(results.best_params_['hidden_layer_sizes']), random_state=1,
batch_size=50, max_iter=200)
clf.fit(X_train, y_train)

# predicting the outcome of the test sample using the optimal model
y_pred_test = clf.predict(X_test)
print('Misclassified samples:%d' % (y_test != y_pred_test).sum())
print('Accuracy Neuaral Network:%.2f' % accuracy_score(y_pred_test,y_test))
```

```
# feature importance and ranking power of the predictors
forest.fit(X_train, y_train )
feat_labels=X_train.columns

# Print the name and gini importance of each feature

for feature in zip(feat_labels, forest.feature_importances_):
 print(feature)

for feature_list_index in sfm.get_support(indices=True):

    print(feat_labels[feature_list_index])
```

Appendix D: SVM Code

```
#preparation of ranking Expected one of: sgd, adam, lbfgs
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.cross_validation import train_test_split
from sklearn.ensemble import RandomForestClassifier

#Loading of data
raw_data = pd.read_excel(r"Modelling data_4.xlsx")
#raw_data = raw_data.fillna(0)

y = raw_data['Bankrupt']
X = raw_data
del X['Bankrupt']

# split into train and test
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)

from sklearn.feature_selection import SelectFromModel
forest=RandomForestClassifier(n_estimators=1000,random_state=0,n_jobs=-1)
sfm = SelectFromModel(forest, threshold=0.0003)
sfm.fit( X_train , y_train )

X_important_train=sfm.transform( X_train  )

X_important_test=sfm.transform(X_test)

sc=StandardScaler()

X_train_std=sc.fit_transform(X_important_train)

X_test_std=sc.fit_transform(X_important_test)

#cross validation and parameter tunin
svm=SVC(kernel='sigmoid',random_state=0,gamma=0.001,C=1000.0)
svm.fit(X_train_std, y_train )
```

```
y_pred= svm .predict(X_test_std)
print('Misclassified samples:%d' % (y_test != y_pred).sum())
print('Accuracy SVM:%.2f' % accuracy_score(y_test,y_pred))

# feature importance
forest.fit(X_train_std, y_train )
feat_labels=X_train.columns

# Print the name and gini importance of each feature

for feature in zip(feat_labels, forest.feature_importances_):
 print(feature)

for feature_list_index in sfm.get_support(indices=True):

    print(feat_labels[feature_list_index])
```