UNIVERSITY OF GOTHENBURG
SCHOOL OF BUSINESS, ECONOMICS AND LAW

Master Degree Project in Economics

The Machines are Coming

Non-parametric methods and bankruptcy prediction - An artificial neural network approach

Ozan Demir

Supervisor: Mattias Sundén
Master Degree Project No. 2016:91
Graduate School

The Machines Are Coming
Non-parametric Methods and Bankruptcy Prediction - An Artificial Neural Network Approach
Ozan Demir

Supervisor: Mattias Sundén, Department of Economics

Master's Thesis 2016
Department of Economics
University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2016

The Machines Are Coming
Non-parametric Methods and Bankruptcy Prediction - An Artificial Neural Network Approach
Ozan Demir
Department of Economics
University of Gothenburg

Abstract

Prediction of corporates bankruptcies is a topic that has gained more importance in the last two decades. Improvement in data accessibility makes the topic of bankruptcy prediction models a widely studied area. This study looks at bankruptcy prediction from a non-parametric perspective, with a focus on artificial neural networks (ANNs). Inspired by the classical work by Altman (1968) this study models bankruptcies with classification techniques. Five different models - ANN, CART, k-NN, LDA and QDA are applied to Swedish, German and French firm level datasets. The study findings suggests the ANN method outperforms other methods with 86.49% prediction accuracy and struggles to separate the smallest companies in the dataset from the defaulted ones. It is also shown that an increase in number of hidden layers from 10 to 100 results in an increase of 1% in prediction accuracy but the effect is non-linear.

Keywords: Bankruptcy prediction, machine learning, non-parametric methods, artificial neural networks.

Contents

List of Figures

List of Tables

1

Introduction

Over the past decade, commercial banks devoted many resources to develop robust internal models to better quantify financial risks and assign economic capital. These ever increasing efforts have been recognized and further encouraged by different regulators. An important question for banks and the regulators is the evaluation of the models predicting accuracy in predicting credit losses. Bankruptcy prediction models for individual obligors are a core part of the assessment made by investors and financial institutions when estimating potential losses. Once a reliable and accurate estimation of the creditworthiness of the firm is made, it is often straightforward to estimate associated losses and loss distributions which can lead to sounder lending/investing decisions. Bankruptcy predictions are also useful from a policy and regulatory perspective, where evaluating systemic risk and performing stress tests on the financial system at both national or global level is a challenge. The latter mentioned use of bankruptcy prediction models has grown in significance after the burst of the financial crisis in 2008. There are however several challenges with estimation of creditworthiness that are owing to limitations of data availability and subjectivity. The subjective factor could be a problem from a consistency perspective and often occurs when the default risk of an obligor is assessed by analysts.

The early credit scoring models were developed by Durand (1941) and Altman (1968) where discriminant analysis is applied to separate creditworthy firms from non-creditworthy firms. Lack of data for the universe of firms led to the development of structural models. The structural models pioneered by Merton (1974) have been popular in both academia and applied finance. The model by Merton (1974) was later developed further by Black and Cox (1976) to increase predictability of default prior to the maturity date of the obligation. Other techniques that have been applied to estimate bankruptcies include regression analysis with Probit (Boyes, 1989) and Logit (Ohlson, 1968) specifications. Many of the above mentioned models have shortcomings related to data-requirements and the ability to address the complexity of the issue. In recent years, some cutting-edge technologies from other disciplines such as Genetic Algorithms (Etemadi et al., 2011) and Neural Networks (NN) (Atiya, 2001; Etemadi et al., 2011; Akkoc, 2012; Sun et al., 2014) have been applied for credit scoring and resulted in better performance than traditional techniques.

To the background that traditional credit scoring techniques such as regressions with probit and logit specifications usually require more assumptions that might not hold and still under-perform compared to some cutting-edge classification techniques, this study aims to model creditworthiness of Swedish corporates by applying different classification techniques on corporate level data. The emphasis will be on non-parametric methods with a focus on artificial neural networks. Five different type of models will be tested, two parametric; linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) and three non-parametric methods - classification and regression tree (CART), k-NN and artificial neural networks (ANN). As a secondary contribution, this study aims to look at the effect of varying hidden layers on the performance of artificial neural networks.

1.1 Credit Risk

The Basel Committee on Banking Supervision defines credit risk as the potential that a borrower or counterpart will fail to meet its obligations in accordance with agreed terms. Assessment of credit risk, and more specifically ensuring accuracy and reliability of the evaluation is of critical importance to many market participants motivated by different objectives. Credit risk management constitutes a critical strategic part for the profitability of a lender. A common way of assessing credit risk is by credit scoring models, in contrast to the subjective opinion of a loan officer, the models analyse the borrowers creditworthiness by looking at quantitative data. A good credit scoring model has to be highly discriminative: high scores reflect almost no risk and low scores correspond to very high risk. An overview of techniques for credit assessment will be presented in the next chapter.

1.1.1 Defining "Creditworthiness"

The creditworthiness of a corporate is a wide concept, it can be either defined as a counterpart that is deemed “creditworthy” of a specific amount of money from a lender and it can also be creditworthy in relative terms, by a specific rating or ranking. This thesis follows a fairly binary perspective on creditworthiness, a company is either creditworthy if it is not defaulted, or it is non-creditworthy if it has defaulted. According to Standard & Poor’s (S&P), a default is first recorded upon the instance of a payment default on a financial obligation. Dividend on stock is not part of a financial obligation that qualifies as a default. In this thesis, a similar approach on default has been followed – a corporate is defaulted if they failed to pay the interest on their loans for more than 90-days, the company is also defaulted of it ceases to exist due to failure of payment of obligations which led to bankruptcy.

1.2 Non-parametric Methods

A non-parametric method in statistics is a method in which no assumption about the functional form of the underlying population distribution is made. Although there are few assumptions made, a common assumption required and made about the objects/observations is that they are independent identically distributed (i.i.d.) from any kind of continuous distribution. As a result of these characteristics or lack of assumptions; non-parametric statistics is also called distribution free statistics.

There are no parameters to be estimated in a non-parametric model. In contrast to non-parametric methods, parametric statistical models are models where the joint distributions of the observations involves unknown parameters, that need to be estimated. In the parametric setting, the functional form of the joint distribution is assumed to be known. Although non-parametric and semi-parametric methods are often lumped together under the title "non-parametric methods", it is worth differentiating between the two. A semi-parametric model is a method that might have parameters but very weak assumptions are made about the actual form of the distributions of the observations.

1.2.1 Non-parametric Density Estimation

In classification applications, the aim is to try to develop a model for predicting a categorical response variable, for one or more predictor (input) variables. In other words, if we know that an observation i.e. creditworthy or non-creditworthy arises from one of different mutually exclusive classes or groups, then more specifically, the aim is to estimate the probability of occurrence in each group at each point in the predictor space. After the estimation of the probabilities, we can assign each estimation point to the class with the highest probability at that point by segmenting the predictor space into regions assigned to the different classes.

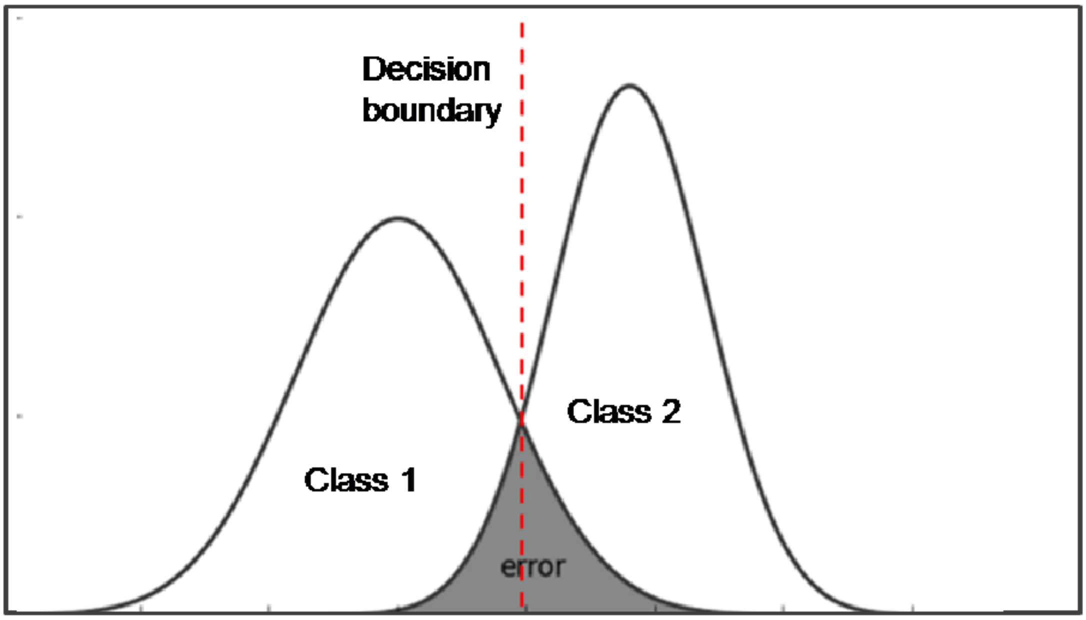


Figure 1.1: Two overlapping distributions and decision boundary.

Both parametric and non-parametric methods can be used to estimate probabilities. The interest is to estimate density function f itself. Let $X_1, X_2, ..., X_n$ be a random sample from a population with unknown probability density function f. If we suppose the random sample is from a distribution with known density function such as Normal distribution with mean μ and σ^2. The density function f can then be estimated by estimating the values of the unknown parameters μ and σ^2 from the sample and substitution the estimates into the function of the normal density. Hence, the parametric density estimator becomes

$$\hat{f}(x) = \frac{1}{\sqrt{2\hat{\sigma}^2}} exp(-\frac{1}{2\hat{\sigma}^2}(x - \hat{\mu})^2) \tag{1.1}$$

where $\hat{\mu} = \frac{\sum_i x_i}{n}$ and $\hat{\sigma}^2 = \frac{\sum_i (x_i - \hat{\mu})^2}{n-1}$.
In case of a non-parametric estimation of a density function, the functional form is not known or not assumed to be known. And there are several methods that can be applied to estimate the density function. One of the oldest density estimator is the *histogram*, where an "origin" x_o and the class width h needs to be specified for the specifications of the following interval

$$I_j = (x_o + j \cdot h, x_0 + (j+1))h \qquad (j = ..., -1, 0, 1, ...) \tag{1.2}$$

for which the number of observations falling in each I_j is counted by the histogram. The choice of "origin" is fairly arbitrary but the role of the class width becomes immediately clear, the form of the histogram highly depends on these two tuning variables. Another popular estimation method is the *kernel estimator* (naive estimator). Similar to the histogram, the relative frequency of observations falling into a small region i computed. The density function f at a point x is as follows

$$f(x) = \lim_{h \to 0} \frac{1}{2h} \Pr[x - h < X \leq x + h]. \tag{1.3}$$

As it is with histograms, the bandwidth "h" needs to be specified however, there is no need for a specification of the origin x_0. Defining the weight function

$$w = \begin{cases} 1/2 \quad if \qquad |x| \leq 1 \\ 0 \quad if \quad otherwise \end{cases} \tag{1.4}$$

Then,

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} w(\frac{x - X_i}{h}) \tag{1.5}$$

Instead of the rectangle weight function $w(\cdot)$ a general, more smooth kernel function $K(\cdot)$ is chosen, the kernel density estimation can be defined as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{1.6}$$

$$K(x) \leq 0, \int_{-\infty}^{\infty} K(x)dx = 1, K(x) = K(-x). \tag{1.7}$$

The estimator depends on the bandwidth $h > 0$. The positivity of the kernel function leads to a positive density estimate $\hat{f}(\cdot)$ and the normalisation $\int K(x)dx = 1$ implies $\int \hat{f}(x)dx = 1$, which is a condition for $\hat{f}(x)$ to be a density.

1.2.2 Why Non-parametric?

The field of credit-assessment has been a popular subject within both parametric and non-parametric methods. In studies such as Hernandez & Torero (2014), the researchers are pointing at the advantage of non-parametric methods instead of parametric for credit scoring when the odds of default are not linear with respect to some of the explanatory variables. Similar results are shown in the paper Kumar & Sahoo (2012) where non-parametric methods outperforms the parametric methods in credit scoring. The advantages of non-parametric methods compared to parametric that are often quoted in literature are as follows.

- Non-parametric methods require less to no assumptions because they don't rely on the assumptions on the shape or parameters of the population distribution.
- Easier to apply on smaller sample sizes
- Can be used on all types of categorical data, which are nominally scaled or are in rank form, as well as interval or ratio-scaled data.
- They are almost as powerful as the parametric method if the assumptions for the parametric methods don't hold and when they hold non-parametric methods generally outperform.

1.3 Statistical Classification and Machine Learning

The term classification occurs in a wide range of topics from social sciences to mathematical statistics. At its broadest meaning, the term covers any context in which some decision or forecast is made on the basis of information available at the time (input variables). Some contexts at which a classification procedure is fundamental are for example mechanical procedures for sorting letters, assigning credit ratings for individuals and corporates based on financial and personal information and even preliminary diagnosis of a disease. The construction of a classification process for a set of data in which true classes are known is also called pattern recognition, discrimination or supervised learning. This is in contrast to the unsupervised learning and clustering in which classes are inferred from data.

Machine learning is in general the term used for automatic computing procedures based on logical or binary operations, that is capable of learning a task by training on input examples. This study is concerned with the classification aspect of machine learning. The aim of machine learning here is to generate classification simple enough for humans to understand. It must mimic human reasoning well enough to both function and provide insight in the decision process of in this case assessing creditworthiness of a company (Weiss Kulikowski, 1991).

There are a wide range of classification techniques used in the field of finance today; one of the oldest classification procedures are the linear discriminants put forth by Fisher (1936), the idea is to divide the sample space by a series of lines. The direction of the line drawn to bisect the classes is determined by the shape of the clusters of the observations. In the case of linear discriminant analysis (LDA), the distributions for each class are assumed to share the same covariance matrix which leads to linear decision boundaries. No assumptions on the covariance matrices lead to quadric discriminant analysis (QDA) allowing for non-linearity in the decision boundary. Other popular techniques are decision trees (referred to as CART in this study), this procedure is based on recursive partitioning of sample space, by dividing the space into boxes and at each stage the boxes are re-examined to determine if there is another split required. The splits are usually parallel to the y- and x-axes(D. Michie, 1994.) Furthermore, the technique that has found more and more applications in classification procedures is the k-Nearest-Neighbour (k-NN), the idea is that it's more likely that observations that are near to each other belong to the same class. The sensitivity of the method can be changed by choosing a proper k (number of nearest neighbours). What is of more importance for this study are the Artificial Neural Networks(ANNs) ,that is finding applications in many different aspects of statistical modelling, including classification. Neural Networks consists of layers of interconnected nodes where every node is producing a non-linear function of the input data that comes from the previous layer (either input data or from previous node). In this sense, the complete network represents a complex set of interdependencies that can incorporate degrees of non-linearity.(Hertz et al. 1991).

The research on optimal performance of classification techniques is not new, both in studies where some of the above mentioned techniques were simulated (Tibshirani LeBlanc, 1992, Ripley, 1994, Buhlmann & Yu, 2003; Kuhnert, Mengersen, & Tesar, 2003;) and in comparative studies in different areas such as business administration (marketing) (Hart, 1992; West et al., 1997) natural sciences (Bailly, Arnaud & Puech, 2007; Liu Chun, 2009) and medicine (Reibnegger, Weiss, Werner-Felmayer, Judmaier, & Wachter, 1991). The major part of the studies looks at the overall percentage of correctly classified cases by different classification techniques and the results are sometimes contradicting. In two different studies such as Ripley (1994) and Dudoits et al (2002) the researchers found that the traditional methods of LDA and QDA performed better than CART but not as good as ANN. In the study by Preatoni et al. (2005) LDA outperformed both CART and ANN but in the contrary, Ture et al. (2005) and Yoon et al. (1993) shows that ANN shows the highest accuracy rate of the above mentioned techniques. West et al.(1997) shows that LDA and CART performs as well as or better than ANN on groups that are linearly separable but in the presence of non-linearity, LDA and CART suffers compared to k-NN and ANN. An overview of classification models is provided in the appendix.

2

Review of Credit Evaluation Models

This chapter will be dedicated to give a brief overview of current credit evaluation models. Credit evaluation models can be divided into groups; structural, statistical and non-parametric. This study focuses on non-parametric evaluation models, however, different perspectives on evaluating the creditworthiness of a company is necessary for better and more complete understanding of the problem that is to be tackled.

2.1 Structural Models

Structural models use the evolution of company's structural assets, such as asset values and debt values to estimate the time of default. Merton (1974) is the first attempt to model the default probability in a structural way.

2.1.1 Merton's Model

In Merton's model, a company defaults if the company's assets are below its outstanding debt at the time of servicing the debt. Merton makes use of the Black and Scholes (1973) option pricing model to build a valuation model for corporate liabilities. This is fairly straightforward when the firm's capital structure and default assumptions are adapted to the requirements of the Black-Scholes model. Assuming the capital structure of the firm is comprised by equity and a zero-coupon bond with maturity T and face value D, of which the value at time t is denoted by E_t and $z(t,T)$ respectively, for $0 \leq t \leq T$. Where the firm's asset value V_t is the sum of equity and debt values. According to these assumptions, the equity can be seen as a call option on the assets of the firm with maturity T and strike price of D. If the firm's asset value V_t at maturity T is equal to or larger than the face value of the debt D then the firm doesn't default. Default happens if $V_t < D$. There are several assumptions that Merton (1974) adopts, these are; firm can only default at time T, in existence of transaction costs, bankruptcy costs, and taxes. There are also assumptions on the lending and the interest rate r, borrowing and lending is unrestricted at a constant r. The value of the firm in Merton's model is invariant when changes in capital structure occurs (Modigliani Miller, 1958).

The firm's asset value is assumed to follow the process

$$dV_t = rV_t dt + \sigma_v V_t dW_t, \tag{2.1}$$

where σ_v is the asset volatility and W_t is a Brownian motion.
The equity- and bondholders pay-offs at time T is given by $max(V_t - D, 0)$ and $V_t - E_t$, respectively

$$E_t = \max[V_t - D, 0] \tag{2.2}$$

$$z(T,T) = V_t - E_t \tag{2.3}$$

Applying the Black-Scholes pricing formula, the value of equity at time $t(0 \leq t \leq T)$ is given by

$$E_t(V_t, \sigma_v, T-t) = e^{-r(T-t)}[e^{r(T-t)}V_t\Phi(d_1) - D\Phi(d_2)], \tag{2.4}$$

where $\Phi(\cdot)$ is the distribution function of a standard normal random variable and d_1 and d_2 are given by

$$d_1 = \frac{ln(\frac{e^{r(T-t)}V_t}{D}) + \frac{1}{2}\sigma_V^2(T-t)}{\sigma_V\sqrt{T-t}}, \tag{2.5}$$

$$d_2 = d_1 - \sigma_V\sqrt{T-t}. \tag{2.6}$$

Then the probability of default at time T is given by

$$P[V_t < D] = \Phi(-d_2). \tag{2.7}$$

An illustration of the model is depicted below.

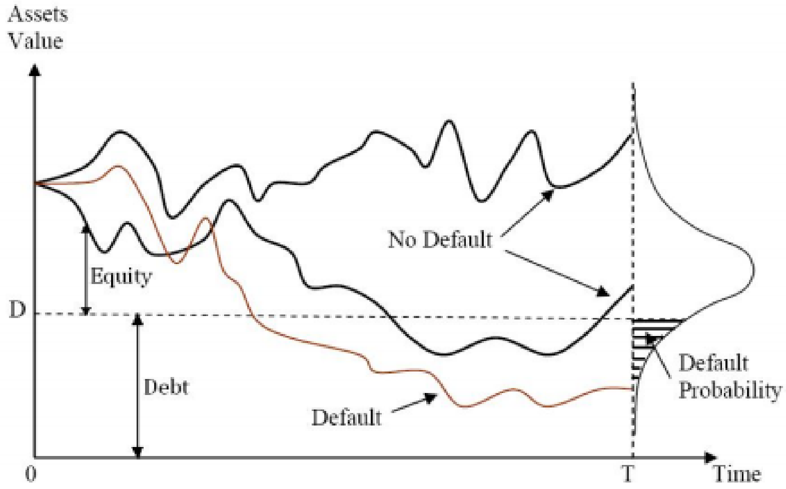


Figure 2.1: Probability of Default in the Merton Model.

Merton's model has many advantages and perhaps one of the most straightforward advantage is that it allows to directly apply the option pricing theory developed by Black and Scholes (1973). There are however necessary assumptions about the asset value process, interest rates and capital structure that needs to be satisfied. As for many financial models, there is a trade-off between realistic assumptions and how easy it is to implement and one could argue that this model ops for the latter. One example of suggestions for improvement is from Jones et al. (1984) where it is argued that introduction of stochastic interest rates and taxes would improve the models performance. The Merton model was later built on by Black and Cox (1976), introducing first passage models and hence making it possible to model the possibility of default at any time t and not only at maturity.

2.2 Statistical Models

A wide range of statistical techniques are applied in credit assessments and credit scoring models. Techniques such as regression analysis, discriminant analysis, probit and logit regression and many more have been applied in the past. This section will be dedicated to give a brief overview of some of the techniques that are relevant for this study.

2.2.1 Discriminant Analysis

The aim of discriminant analysis is to find a discriminant function to classify objects (in this case, creditworthy and non-creditworthy) into two or more groups, based on a set of features that characterizes the objects. In another words, the techniques aim to maximize the difference between the groups, while trying to minimize the differences among members of the same group.

Discriminant analysis application on credit scoring is first attempted by Durand (1941), with a linear discriminant analysis Model (LDA). This model was further developed by using company specific data in Beaver (1966), Altman (1968), Meyer and Pifer (1970), Sinkey (1975), Martin (1977), West (1985) and many others. In the important work of Altman (1968) a classical multivariate discriminant analysis technique (MDA) is used, which builds on the Bayes classification procedure. It assumes that the two classes (default and non-default) have Gaussian distributions with equal covariance matrices. These assumptions and the ability of the method to justify these assumptions are criticized in Thomas (2000) and West (2000). The following financial ratios were used by Altman (1968) as inputs in Altman's Z-score model.

Financial Ratio
working capital /total assets
retained earnings /total assets
EBITA/total debt
market capitalization /total debt
sales/total assets

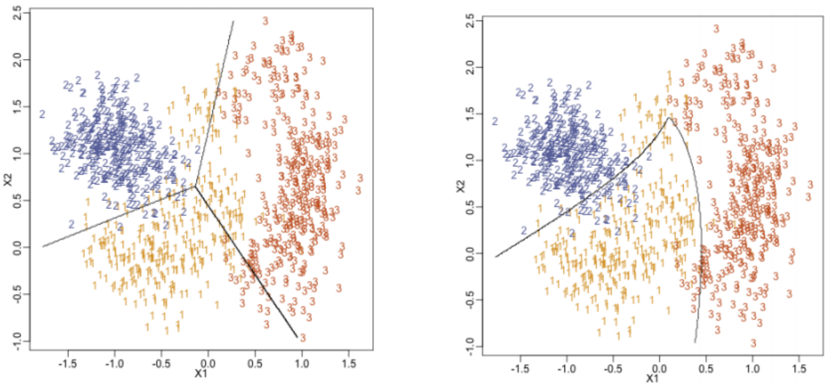
Table 2.1: Altman's Ratios

The model is using the following discriminant function to classify the companies into groups

$$Z = \lambda_1 x_1 + \lambda_2 x_2 + ... + \lambda_n x_n, \tag{2.8}$$

where x_i represents inputs used (see above) as independent variables and λ_i indicates discrimination coefficients.

Discriminant analysis can be divided into different categories with different strengths and weaknesses. This study will focus on linear discriminant analysis (Applied by Altman) and a generalization of the that method; Quadratic Discriminant Analysis(QDA). LDA generally needs fewer parameters to estimate the discriminant function, however it is inflexible and can struggle to separate groups with different underlying distributions. The lack of assumption on the distribution of the groups makes QDA more flexible but it might also be less accurate compared to LDA. Below, an example of LDA and QDA is presented.



(a) LDA decision boundaries **(b)** QDA decision boundaries

LDA and QDA are two very common techniques applied in credit scoring, Myers and Fogly (1963) compares regression models to discriminant analysis. West (2000), Abdou Pointon (2009) and Gurny Gurny (2010) compare the predictive power of probit, logit and discriminant analysis in different settings and find that the Logit and Probit specifications outperforms DA (both LDA and QDA) and the Logit-model outperforms both the Probit and the DA. Similar results can be seen in Guillen Artis (1992), where Probit outperforms DA and linear regression models.

2.2.2 Regression Models

Probit and logit regression analysis are two multivariate techniques that are in this case used to estimate the probability that default occurs by predicting a binary dependent variable from a set of independent variables. The response (binary dependent outcome) y_i, is equal to 0 if default occurs (with probability P_i) and 1 if default does not occur (with probability 1-P_i). Assume the following model specification to estimate the probability P_i that default will occur

$$P_i = f(\alpha + \beta' x_i), \tag{2.9}$$

where x_i are financial indicators and α and β are estimated parameters.
Two of the many ways of specifying P_i, namely probit and logit transformation are as follows.

2.2.2.1 Probit Model

The probit analysis is a widely used regression method in credit assessment for both personal loans and corporate credits. The methodology was pioneered by Finney (1952) in toxicological problems and the first applications of probit models on corporate default prediction is seen in Altman et.al (1981) and Boyes (1989). In the case of probit model the cumulative distribution function of a normal distribution is as follows

$$P_i = \int_{-\infty}^{\alpha+\beta' x_i} \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}t^2)dt. \tag{2.10}$$

2.2.2.2 Logit Model

The logistic regression (LR) approach to estimating default probabilities was introduced by Ohlson (1968) and later further developed by Chesser (1974), Srinivasan Kim (1986), Steenackers Goovaerts (1989) applies the logit model to personal loans,, the LR has been widely used in both research and practice for PD estimation (Aziz et al.,1988 ;Gentry et al., 1985; Foreman, 2003; Tseng and Lin, 2005) and has the following specification

$$P_i = \frac{exp(\alpha + \beta' x_i)}{1 + exp(\alpha + \beta' x_i)} = \frac{1}{1 + exp(-\alpha - \beta' x_i)}. \tag{2.11}$$

Because of the non-linear features of these models, it is necessary to use maximum likelihood estimation. The likelihood function would be defined as

$$L = \prod_{i=1}^{N} Pr(y_i = 1 \mid X_{it}, \beta, \alpha)^{y_i} Pr(y_i = 0 \mid X_{it}, \beta, \alpha)^{1-y_i}. \tag{2.12}$$

The LR model does not necessarily require the same assumptions as LDA or MDA (see below) but Harrell and Lee (1985) shows that the LR performs better than LDA even though the necessary assumptions for LDA are fulfilled.

2.3 Non-Parametric Methods

Below, the relevant non-parametric methods for this thesis will be presented. A more detailed presentation of the non-parametric method artificial neural networks will be presented in the coming chapter. In contrast to the above mentioned methods, the non-parametric methods usually requires less assumptions. The methods look at the default probability estimation from a classification perspective, this view translates into classifying credits into bad or good credits, where the default probability is the determinant of the credits quality.

2.3.1 CART Models

Decision trees or Classification and Regression Trees (CART) are classification techniques that have been widely applied in credit assessment techniques. The CART model was pioneered by Breiman et al. (1984) although it was earlier stated in Raiffa Schlaifer (1961) and Sparks (1972). Early attempts to use CART in a credit scoring application can be seen in Frydman, Altman and Kao (1985) and Makowski (1985). CART is a non-parametric method for predicting continuous dependent variables and categorical predictor variables. The method employs binary trees and classifies observations into a number of classes. The basic idea of the decision tree is to split the given dataset into subsets by recursive portioning. The splitting points (attribute variables) are chosen based on Gini impurity and the Gini gain is given by

$$i(t) = 1 - \sum_{i=1}^{m} f(t,i)^2 = \sum_{i} f(t,i)(t,j), \tag{2.13}$$

$$\Delta i(s,t) = i(t) - P_L i(t_L) - P_R i(t_R) \tag{2.14}$$

Where $f(t,i)$ is the probability of obtaining i in node t, the target variables takes values in $\{1,2,3,...,m\}$. P_L is the proportion of cases in node t divided to the left child node and P_R is the proportion of cases in t sent to the right child node (see nodes and "child nodes in the figure below). If there is no additional Gini gain, or the stopping rule is satisfied, the splitting process stops and a decision tree with nodes and cut-off values are created. The figures below illustrates an example of application of the CART algorithm.

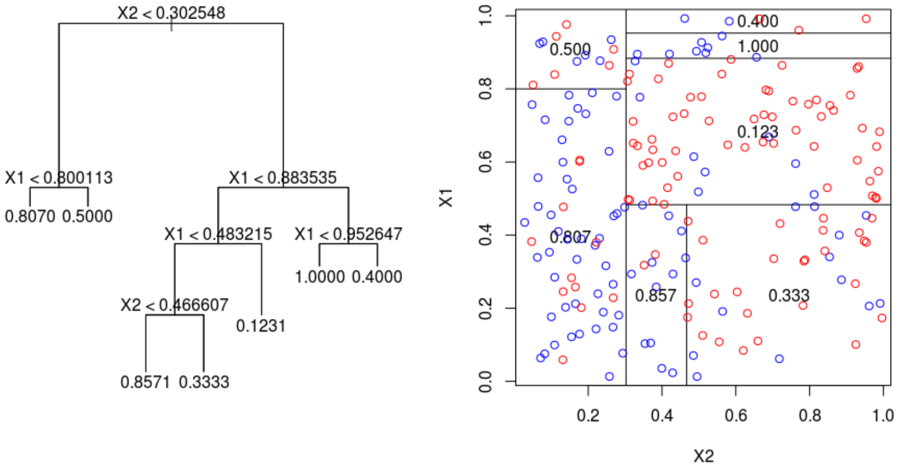


Figure 2.2: Example of an CART application.

CART has been compared to other methods in several studies, Frydman, Altman and Kao (1985) Coffman (1986) , Boyle et al. (1992) shows that CART outperforms DA. The type of method used can affect the choice of explanatory variable, Devaney (1994) compares CART to logistic regression and finds that these models select different financial ratios as explanatory variables for default prediction.

2.3.2 k-Nearest Neighbour

The k-Nearest Neighbour (k-NN) method is a non-parametric method use for many purposes such as probability density function estimation and classification (clustering) technique. It was first proposed by Fix and Hodges (1952) and Cover and Hart (1967). There are several reasons for why it was chosen as a suitable method for credit scoring and bankruptcy prediction problems:

1. The non-parametric nature of the k-NN method makes it possible to model irregularities over feature space.

2. According to Terrel and Scott (1992) the k-NN method has been found to perform better than other non-parametric methods when the data are multidimensional.

3. The k-NN method is relatively intuitive and can be easily explained to managers who needs to approve its implementation.

The k-NN method aims to estimate the good or bad risk probabilities (creditworthy or non-creditworthy) for a company to be classified by the proportions "good" or "bad" among the k "most similar" points in a training sample. The density function estimation is very similar to the kernel estimation described above where the density estimation function is defined as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{2.15}$$

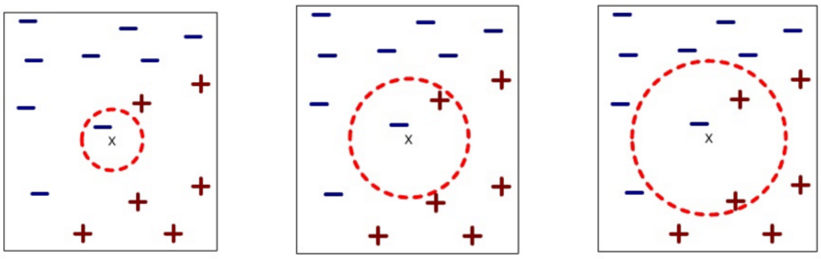
$$K(x) \geq 0, \int_{-\infty}^{\infty} K(x)dx = 1, K(x) = K(-x). \tag{2.16}$$

K is a pre-defined kernel (Gaussian and Epanechnikov among the most popular ones). The bandwidth h is also called the "smoothing parameter", in other words when $h \to 0$ the distribution is getting "spikes" at every observation X_i and $f(\cdot)$ becomes more smooth as h increases. The k-NN estimation is different from the kernel-estimation in the bandwidth selection, instead of using a global bandwidth, a locally variable bandwidth $h(x)$ can be chosen. The idea is to use large bandwidth for regions where the data is more sparse. In other words

$$h(x) = \text{A chosen distance metric of } x \text{ to the kth nearest observation}$$

where k is determining the magnitude of the bandwidth.

The precision of the algorithm depends both on the distance metric and the number of neighbours that are pre-defined. In the illustrations below, there are three different classifications with three different k presented.



(a) 1-Nearest Neighbour **(b)** 2-Nearest Neighbour **(c)** 3-Nearest Neighbour

Another key element in this classification technique is the similarity of the points, it is assessed using different distance metrics.

2.3.2.1 Distance Metrics

An appropriate distance measure is a critical feature of the k-NN method. The aim of the selection process is to choose a metric to improve the performance of the classification according to some pre-specified criterion. The conventional approach is concentrated on the NN rule and has minimization of the difference between finite sample misclassification rate with the asymptotic misclassification rate as the performance criterion.
A common distance measure is the Euclidean metric given by

$$d_1(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})} \tag{2.17}$$

where $\mathbf{x}$ and $\mathbf{y}$ are points in feature space. However, d_1 may not always be the most appropriate distance measure to use. Fukanaga and Flick (1984) considered the problem of selecting data-dependent versions of the Euclidean metric. They introduced a general approach for incorporating information from the data through the following metric

$$d_2(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{A} (\mathbf{x} - \mathbf{y})} \tag{2.18}$$

where $\mathbf{A}$ can be any symmetric positive define matrix. Local metrics are defined as those for which $\mathbf{A}$ can vary with $\mathbf{x}$ and global metrics in contrary are those for which $\mathbf{A}$ are independent of $\mathbf{x}$. In the case of the global metrics, the distance between two points depends only on their relative position. Fukanaga and Flick (1984) suggests using a global metric for mean-squared error minimizations. The risk of using a local metric approach is that the metric might incorporate local information or features of the training set that simply are not representative for the population. Because the metric in the local case must be determined from a small region around $\mathbf{x}$, it can be difficult to determine the metric accurately. D. Michie et al (1994) for other examples of distance metrics.

2.3.3 Cutting-edge Non-parametric Classification Models

The area of classification is developing with a fast pace due to the demand for new methods to process newly available big-datasets. Some of them build on previous methods while others are techniques from other fields that finds applications in mathematical modelling. Below are two of many cutting-edge techniques for classification, although they are not in focus in this study, it is worth mentioning the basic ideas and performance of them. Two modern non-parametric classification techniques - Support vector machines (SVMs) and Genetic Algorithm (GA) are briefly touched upon below. Artificial neural networks (ANNs) which also falls into this category will be presented more thoroughly in the next chapter.

2.3.3.1 Support Vector Machines

Support vector machines (SVMs) were introduced by Boser, Guyon Vapnik (1992) and developed into a rather popular method for binary classification. The method found application in a range of problems, including pattern recognition (Pontil Verri, 1998), text categorization (Joachims, 1998) and credit scoring (Huang et al, 2007 , Besens, 2003, Li, 2004). The basic idea (as it is in many other classification methods) is to find a hyperplane that correctly separates the d-dimensional data into two classes. Since sample data is not always (rarely) linearly separable, SVMs tackles the problem by casting the data in a higher dimensional space, where the data is often separable. However, higher-dimensions typically comes with computational problems, one of the key insights used in SVMs is the way it deals with higher-dimensions, as a result it eliminates the above mentioned concern. The non-linear casting of the data into higher dimensions is defined in terms of kernel function. In other words, SVM can in general be understood as an application of linear technique in a feature space that is obtained by non-linear preprocessing (Christianini and Shawe-Taylor, 2000).

In comparative studies, the SVM is often compared to ANNs, GAs and CART, in a such a study, Huang et Al (2007), shows that SVM performs slightly better than the others on Australian credit data. Similarly Li (2004) and Schebesch Stecking (2005) shows that SVMs performs slightly better than other techniques when credit scoring Chinese and German data respectively. Baesens (2003) in a similar study finds that SVMs performs well but not as good as ANNs.

2.3.3.2 Genetic Algorithm

Another non-parametric method that wont be in the scope of this thesis but has been applied extensively in recent years is the Genetic Algorithm. Most of the applied techniques being extensions of genetic algorithms by Golgberg (1989).and Koza (1992). Genetic algorithms (GA) are developed to solve non-linear, non-convex global optimization problems by mimicking Darwin principles of Darwinian natural selection and was pioneered by Holland (1975). The GA's have been traditionally used in optimization problems as stochastic search techniques in large and complicated spaces. In recent years been applied to overcome some o the shortcomings in existing models of PD estimation. One major difference between GAs and other non-linear optimization techniques is that they search by maintaining a population of solutions from which better solutions are created instead of making incremental changes to one solutions of a problem (Min et al., 2006). In a GA, a population of strings (called chromosomes) which encode, a potential solution to the problem (called individuals) , is evolved toward a better solution by building on the previous individuals until it is at levels regarded as optimal. In the case of credit scoring, the GAs are used to find a set of defaulting rules based on the cut-off value of several selected financial ratios (Bauer, 1994 and Shin and Lee 2002). One example of genetic algorithm applied to credit scoring is Gordini (2014), where GAs are applied to find cut-offs for each of the pre-determined financial ratios at which the company is considered bankrupt. Similar to a CART tree the genetic algorithm produces a GP tree, the representation of a tree can be explained based on "function" and "terminal" sets where the function set represents simple mathematical operators ($+, -$,x, $\div$) and conditional statements (if... Then...) and the terminals contains inputs, equations etc. A representation is depicted below.

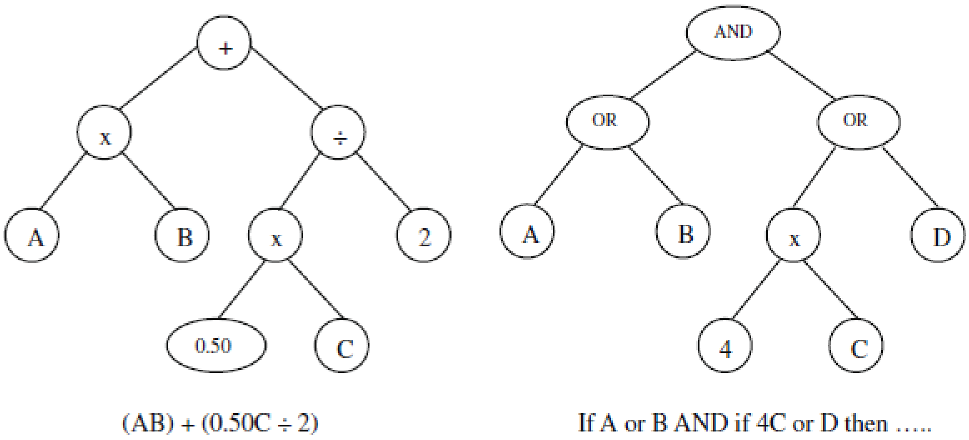


Figure 2.3: Examples of GP trees using simple mathematical operators and conditional statements

Many comparative studies have been conducted on the ability of GAs to outperform other methods in PD estimations. Rafiei et al. (2011) finds that GAs lower accuracy rates than Neural Networks (NN), Etemadi et al. (2009) compares GAs to MDA by applying the methods on an Iranian dataset and shows that GAs outperform MDA.

3

Artificial Neural Networks

This section will be dedicated to present the theory of artificial neural networks. Artificial neural networks is a technique that finds applications in diverse fields, such as character recognition, stock market prediction, machine learning and many more. The technology is inspired and motivated by human brains character and ability to process highly complex, non-linear and parallel systems. The brain has the capacity to organize and structure its essential components called neurons to perform computations such as pattern recognition, perception, etc. A neuron is fundamental to the neural network in the sense that it is the unit that is processing the information. Neural networks as computing machines was first introduced by McCulloch and Pitts (1943) and the first rule of self-organized learning was postulated by Hebb (1949).

An artificial neural network could be thought of as a machine that is designed to mimic and perform tasks similar to the human brain. The neural network is built up by units, that are often represented of nodes and connected to each other through synapses.

3.1 Building Blocks of An Artificial Network

3.1.1 Artificial Neurons

Neurons are the computing and information-processing unit of a neural network, the four fundamental and basic elements of the artificial neuron are as follows

1. *Synapses* or *connecting links*, where each of them are characterized by a *weight* or *strength.* More specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the weight of the synapse w_k.

2. An *adder* or *linear combiner* that is summing the input signals, weighted by the respective synaptic weight of the neuron.

3. An *activation function* for limiting the amplitude of the output of the neuron. The typical amplitude range of the output of the neuron is $[0, 1]$ or $[-1, 1]$.

4. *Bias, b_k* The model of a neuron includes an external *bias*, b_k which

can have an increasing or decreasing effect on the input of the *activation function.*

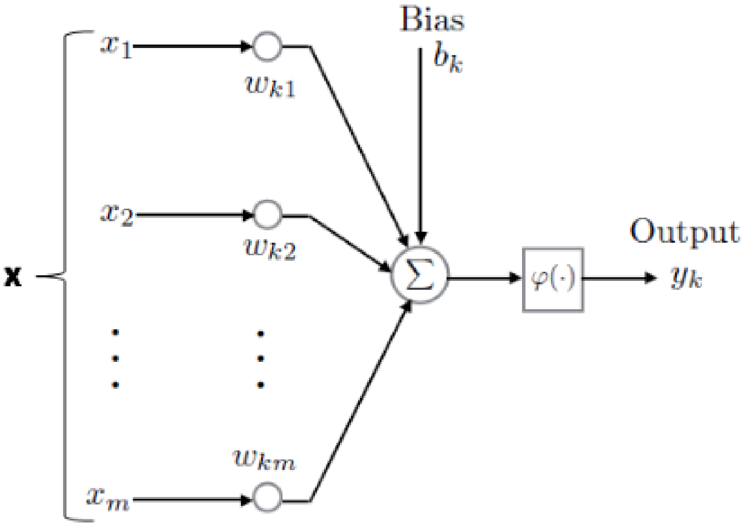


Figure 3.1: Example of a nonlinear model of a neuron.

Figure 3.1 includes a bias b_k, this bias has the effect of increasing or lowering the net input of the activation function.

The neuron k can be described in mathematical terms by the following equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

and

$$y_k = \varphi(u_k + b_k) \tag{3.2}$$

where $w_{k1}, w_{k2}, ..., w_{km}$ are the weights of the synapses of the neuron k and $x_1, x_2, ..., x_m$ are inputs, y_k is the output of the neuron, u_k is the *linear combiner output*, b_k is the bias and $\varphi(\cdot)$ is the *activation function.* The linear combiner and the bias' effect on the output is given by

$$v_k = u_k + b_k \tag{3.3}$$

where the bias b_k can be either positive or negative and is a related transformation to the output u_k of the linear combiner. The *activation potential* or *induced local field* v_k of neuron k is defined as

$$v_k = u_k + b_k. \tag{3.4}$$

Equivalently, the combination of the above mentioned equations can be formulated as follows:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{3.5}$$

and

$$y_k = \varphi(v_k) \tag{3.6}$$

Where $\varphi(v_k)$ is the *activation function.* To account for the external parameter b_k which is the bias, a new synapse is added with the input

$$x_0 = +1 \tag{3.7}$$

and the weight of that synapse is

$$w_{k0} = b_k. \tag{3.8}$$

Hence, the external parameter is controlled for by (1) adding a new input signal and (2) adding a synaptic weight equal to the bias b_k.

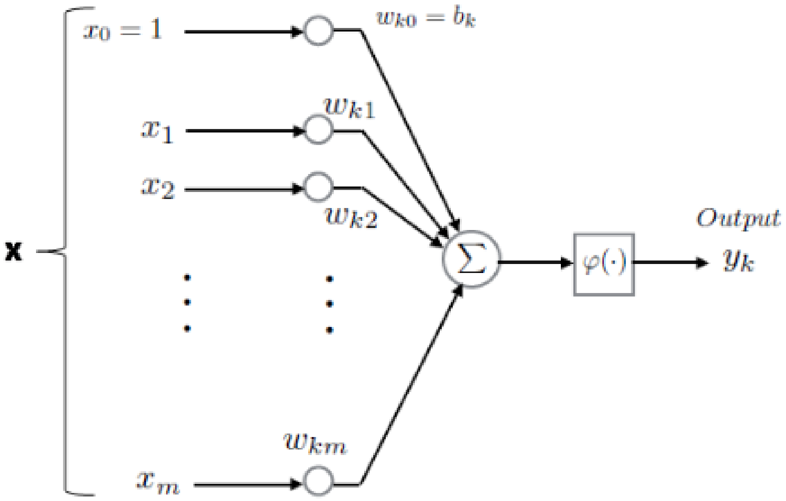


Figure 3.2: Example of a Neural Network with bias b_k as input.

3.1.1.1 Activation functions

The output of the neuron in terms of the *induced local field* v is denoted by the activation function $\varphi(v)$. There are several types of activation functions with different characteristics, below, the three basic types are presented.

1. **Threshold/Heaviside Function**. Is given by

$$\varphi(v) = \begin{cases} 1 & if \quad v \geq 0 \\ 0 & if \quad v < 0 \end{cases} \tag{3.9}$$

Output y_k of a neuron k employing such a threshold is given by

$$y_k = \begin{cases} 1 & if \quad v_k \geq 0 \\ 0 & if \quad v_k < 0 \end{cases} \tag{3.10}$$

where v_k is the induced local field of the neuron k and is given by

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{3.11}$$

This neuron has an *all-or-none property*, the output takes values 1 of v_k is nonnegative, and 0 otherwise. It is pioneered by McCulloch and Piits (1943) and is often referred to as *the McCulloch-Pitts model.*

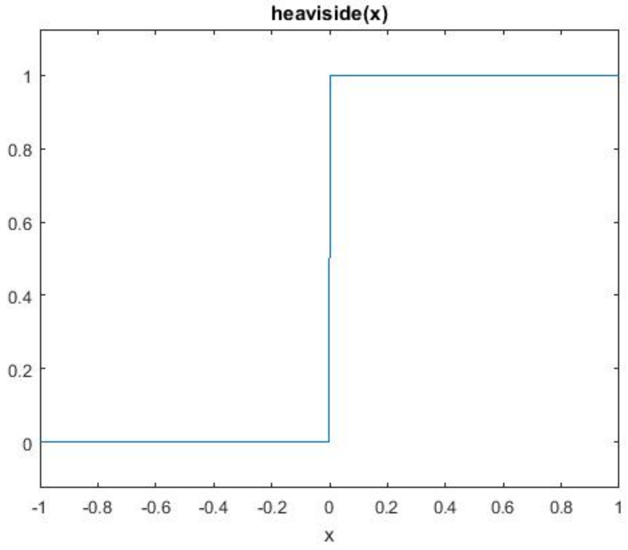


Figure 3.3: Threshold function.

2. **Sigmoid Function**. Is one of the most common of the activation functions in artificial neural networks. In contrast to the threshold function that assumes the value of 0 or 1, the sigmoid function assumes a continuous range of values between 0 and 1. The multilayer perceptron especially requires $\varphi(\cdot)$ to be continuous, differentiability is the key requirement that an activation function has to satisfy for many types of ANN's. An example of nonlinear activation function that is continuously differentiable are sigmoid functions. Two different forms of these functions are:
1.1 *Logistic Function*, in its general form it is defined by

$$\varphi_j(v_j(n)) = \frac{1}{1 + exp(-av_j(n))} \qquad a > 0 \text{ and } -\infty < v_j(n) < \infty \tag{3.12}$$

where $v_j(n)$ is the *induced local field* of neuron j and a is the slope parameters of the function that can be changed to obtain functions with different slopes, see figure for illustration.

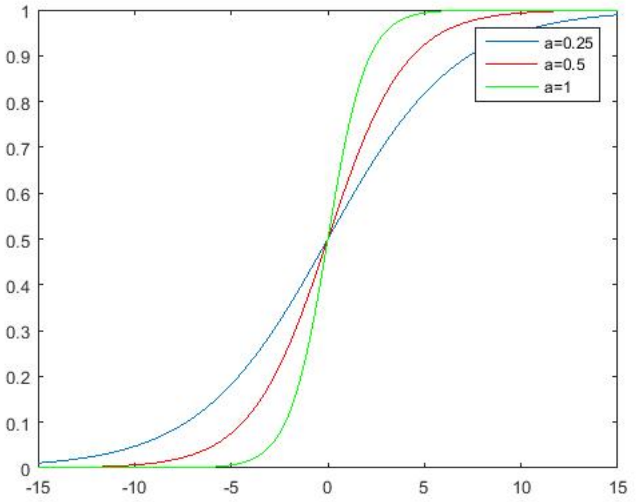


Figure 3.4: Sigmoid function with different a.

1.2 *Hyperbolic tangent function* One of the other commonly used sigmoid functions is the hyperbolic tangent functions. In its general form, it is defined by

$$\varphi_j(v_j(n)) = a \tanh(bv_j(n)), \qquad (a, b) > 0 \tag{3.13}$$

where a and b are constants. The hyperbolic tangent function can be seen as the rescaled and biased version of logistic function.

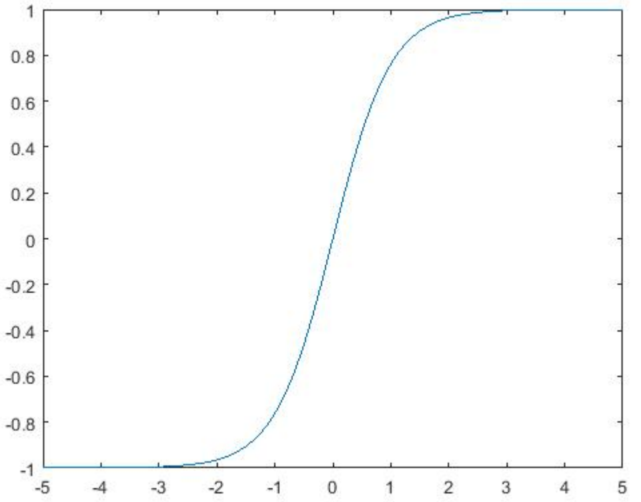


Figure 3.5: Hyperbolic Tangent Function.

3.2 Network Architectures

In this section, two of the most common architectures (structures) of ANN's will be presented, although it is possible to identify three fundamentally different types of network architectures. The *single-layer network*, is identified as a single input layer of nodes at which information flows forward to the output layer of neurons, whereas the *multilayer network* can be constructed of more than one hidden layers. The third type of architecture is different in the way information flows through the network, the *recurrent network* has at least one feedback loop at which the information can flow back to previous nodes(Haykin, 2009).

3.2.1 Single-Layer Feedforward Neural Networks

In the simplest form of a neural network that is layered, there exists an input layer of source node that projects onto the output-layer of neurons (also called computation nodes) but not vice-versa, hence the feedforward attribute. In other words, the network is strictly acyclic, which means information is flowing in one direction. An example of a single-layer feedforward network is illustrated below.

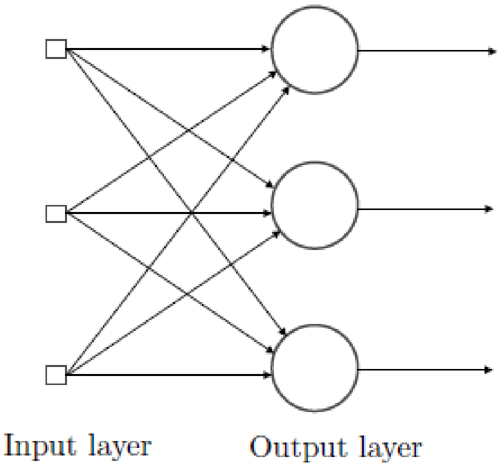


Figure 3.6: Example of a single-layer feedforward network.

3.2.1.1 Perceptron

The perceptron is the simplest form of a neural network used for classification of patterns that are linearly separable (see figure). It was pioneered by Rosenblatt (1958) around the McCulloch-Pitts (1943) non-linear model of neuron. The goal of the perceptron is to accurately and correctly classify the set of externally given input $x_1, x_2, ..., x_m$ into one of the classes φ_1 or φ_2. The classification works through a decision rule that assigns the inputs $x_1, x_2, ..., x_m$ to class φ_1 if the perceptron output is $+1$ and to class φ_2 if output is -1. Figure shows an illustration of a map of the decision regions in the m-dimensional signal space. Where the two regions are separated by a *hyperplane* defined by

$$\sum_{i=1}^{m} w_i x_i + b = 0 \tag{3.14}$$

The *synaptic weights* $w_1, w_2, ..., w_3$ of a perceptron can be adapted by an error-correction rule adapted on an iteration-by-iteration basis , that is known as the perceptron convergence algorithm.

If we treat the bias b_n, as a synaptic weight that is driven by a fixed input equal to $+1$, the $(m+1)$-by-1 may then be defined as the input vector

$$x(n) = [+1, x_1(n), x_2(n), ..., x_m(n)]^T$$

(3.15)

where the iteration step applied to the algorithm is denoted by n. The $(m+1)$-by-1 weight vector is defined as

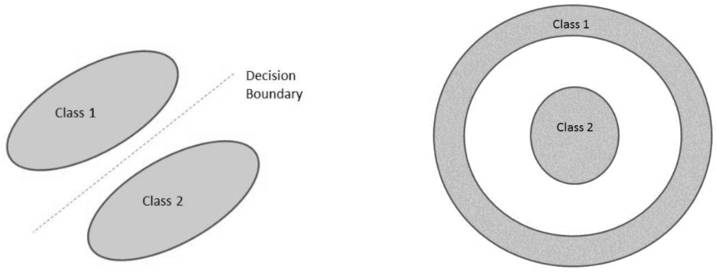
$$w(n) = [b(n), w_1(n), w_2(n), ..., w_m(n)]^T$$

. (3.16)

The linear combiner output is given by

$$v(n) = \sum_{i=0}^{m} w_i(n) x_i(n) = W^T(n) x(n) \tag{3.17}$$

where $w_0(n)$ represents the bias. Suppose that the input variables belong to two linearly separable classes (see figure). Let Ψ_1 be the subset of training vectors $x_1(1), x_1(2), ..$ that belongs to the specific class φ_1 and consequently, let Ψ_2 be the subset of training vectors $x_2(1), x_2(2), ...$ that belongs to the class φ_2. The complete training set is the union Ψ.



(a) Example of linearly separable classes

(b) Example of linearly non-separable classes

Figure 3.7: Examples of linearly separable and non-separable clusters.

The training process involves the separation of the two classes φ_1 and φ_2 by adjusting the weight vector w. Hence, there exists a weight vector w that can be stated as $w^T x > 0$ for every input vector x that belongs to class φ_1 and $w^T x \leq 0$ for every input vector x that belongs to class φ_2. To solve the classification problem the perceptron will find a weight vector w such that the inequalities of the equations above are satisfied. The weight vector of the perceptron is estimated and adapted by the following procedure.

1. No correction is made to the weight vector of the perceptron if the nth input of the training set, $x(n)$ is correctly classified by the vector $w(n)$ that is computed at the nth iteration of the algorithm according to the following rule:

$$w(n+1) = w(n) \qquad \text{if } w^T x(n) > 0 \text{ and } x(n) \text{ belongs to class } \varphi_1 \tag{3.18}$$

$$w(n+1) = w(n) \qquad \text{if } w^T x(n) \leq 0 and\ x(n) \text{ belongs to class } \varphi_2 \tag{3.19}$$

2. If that is not the case, the weight vector w of the perceptron is updated according to the following rule

$$w(n+1) = w(n) - \eta(n)x(n) \quad if\ w^T(n)x(n) > 0 \text{ and } x(n) \text{ belongs to class } \varphi_2 \tag{3.20}$$

$$w(n+1) = w(n) + \eta(n)x(n) \quad if\ w^T(n)x(n) \leq 0 \text{ and } x(n) \text{ belongs to class } \varphi_1 \tag{3.21}$$

where parameter $\eta(n)$ which is the learning-rate parameter that controls the adjustment made on the weight vector w at iteration n.

The theorem that states the convergence of the fixed increment adaptation rule at $\eta = 1$ is as follows. The value of η is unimportant as long as it is positive.

Theorem 3.1 1. *If $\eta(n) = \eta > 0$, where η is a constant independent interation number n, then there exists a fixed increment adaption rule for the perception.*

Proof. See Haykin (2009) □

Now consider the absolute *error-correction procedure* for adapting the single-layer perceptron. In this procedure, $\eta(n)$ is variable. assume $\eta(n)$ is the smallest integer for which

$$\eta(n)\mathbf{x}^T(n)\mathbf{x}(n) > \mid \mathbf{w}^T(n)\mathbf{x}(n) \mid \tag{3.22}$$

with this procedure it can be found that if the inner product $\mathbf{w}^T(n)\mathbf{x}(n)$ at iteration n contains an incorrect sign, then, $\mathbf{w}^T(n+1)\mathbf{x}(n)$ at iteration $n+1$ will have the correct sign. Hence, by setting $\mathbf{x}(n+1) = \mathbf{x}(n)$ one can modify the training sequence at iteration $n+1$ if $\mathbf{w}^T(n)\mathbf{x}(n)$ has wrong sign. This means each pattern is repeatedly presented to the perception until the presented pattern is classified correctly. It is also important to note that using an initial condition different from $\mathbf{w}(0) = (0)$ does not significantly affect the number of iteration required to converge. The convergence of the perceptron is hence assured regardless of the value that is assigned to $\mathbf{w}(0)$. To this background, by using **Theorem 3.1.1.** we can state the fixed increment convergence theorem from Rosenblatt (1962):

Theorem 3.2 1. *If there exists two linearly separable subsets of training vectors φ_1 and φ_2 and the these vectors are inputs to the perceptron. Then the perceptron convergence after some n_0 iterations. In the sense that*

$$\boldsymbol{w}(n_0) = \boldsymbol{w}(n_0 + 1) = \boldsymbol{w}(n_0 + 2) = ... \tag{3.23}$$

is a slution vector for $n_0 \leq n_{max}$

Proof. See Haykin (2009)

3.2.2 Multilayer Neural Networks

Another form of a neural network is the *multilayer neural network*, it distinguishes itself from the single layer structure by the existence of one or more hidden layers, that consists of computation nodes. The addition of one or more hidden layers makes it able for the network to extract higher-order statistics Haykin (2009). This new attribute of the network is highly valuable when the size of the input layer is particularly large.

The information that is supplied to the network through the input layer (source nodes) passes through the first hidden layer of nodes and the output signals of the first hidden layer are input for the second hidden layer on so on to the rest of the network.

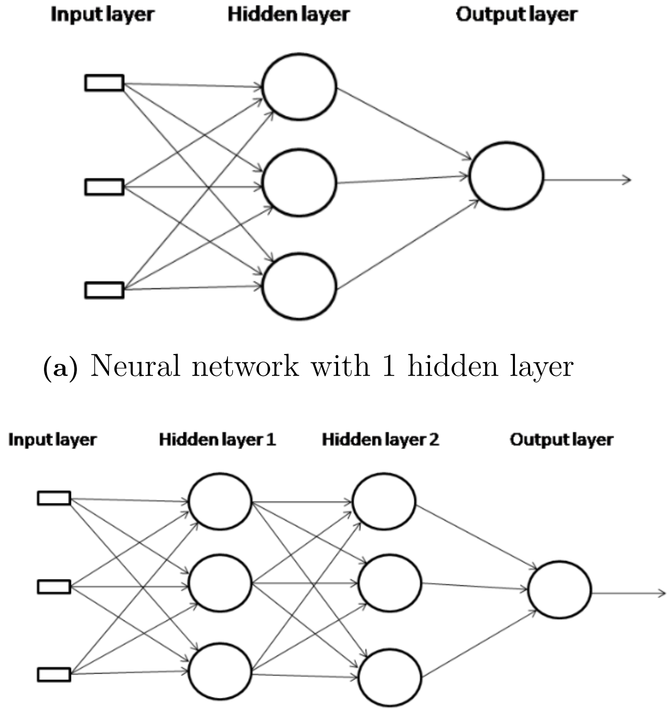
3.2.2.1 Multilayer perceptron

The multilayer perceptron (MLP) has been applied and successfully solved several sorts of difficult and diverse problems(Haykin,2005). The MLP is different from the single layer perceptron in several ways, perhaps the more distinctive difference is the absent of hidden layers in the single layer perceptron but also the requirement for a differentiable activation function in the MLP.The three distinctive characteristics of a multilayer perceptron are as follows:

1. The network includes neurons that has non-linear activation function, it is important that the non-linearity is smooth and differentiable everywhere. Examples of differentiable and non-linear activation functions are given in the previous section.

2. The network contains either one or more layers of hidden neurons,these hidden neurons makes it possible for the network to learn complex tasks by progressively extract more meaningful features from the input signals that flows into them.

3. There is high degrees of connectivity that is determined by the synapses of the network.



(a) Neural network with 1 hidden layer

(b) Neural Network with 2 hidden layers

Figure 3.8: Examples of Multilayer perceptrons

Figures above shows the architectural graph of two different multilayer perceptrons, with one and two hidden layers, the networks illustrated are fully connected. The fully connected character means that a neuron in any layer of the structure is connected to every previous neurons in the previous layer.

The MLP derives its computation power from the above mentioned characteristics and the ability to learn from experience through training. In a general sense, the function of a hidden layer of neurons is to intervene between the input variables (signals) and the network output in an computational way. More specifically every hidden layer performs computations of the function signal appear at the output of each neuron and estimate the gradient vector.

A MLP that is trained with any form of method can be seen as a method of a non-linear input-output mapping. If a continuous and differentiable function such as the logistic function is used then a solution to the above explain context is embodied in the *universal approximation theorem*. The theorem is directly applicable to the MLP and can be stated as:

Theorem 3.3 1. *Let $\varphi(\cdot)$ be a bounded, nonconstrant, and monotone-increasing-continuous function. Let I_{m_0} represent the m_0-dimensional unit hypercube $[0,1]^{m_0}$. The space of continuous functions on I_{m_0} is denoted by C(I_{m_0}). In that case, given any function C(I_{m_0})$\in f$ and $\varepsilon > 0$, there exist an integer M and sets of real constants $a_i, b_i and w_{ij}$ where $i = 1, ..., m_0$ such that the following can be defined*

$$F(x_1, ..., x_{m_0}) = \sum_{i=1}^{m_1} \alpha\varphi(\sum_{i=1}^{m_0} w_{ij}x_j + b_i) \tag{3.24}$$

as an approximation of the function $f(\cdot)$, that is,

$$(x_1, ..., x_{m_0}) - f(x_1, ..., x_{m_0}) \mid < \epsilon \tag{3.25}$$

for all $x_1, x_2, ..., x_{m_0}$ that lie in the input space.

Proof. See Haykin (2009). □

For a better the application of the universal approximation theorem on MLP, an output of a multilayer perceptron stated by Eq.(3.43) can be seen as neural networks has m_0 input nodes with inputs $x_1, ..., x_{m_0}$ and m_1 neurons in a single hidden layers. The hidden neuron i has weights $w_{i_1}, ..., w_{m_0}$ and a bias b_i. And the output is a linear combination of output from hidden layers with synaptic weights of the output layer $a_1, ..., a_m$.

Hence, the universal approximation theorem makes it possible under certain assumptions to use neural networks for function approximation.

3.2.2.2 Optimal number of hidden neurons

The optimal number of neurons is a problem that is often faced in the practical implementation of MLP. The problem more formerly is to find the number of hidden neurons m_1, where the criterion often used is the smallest number of hidden neurons that results in a performance (probability of correct classification) as close to the *Bayesian classifier* as possible, see Haykin (2009) for more information on the Bayesian classifier. After the convergence of a network that has been trained with a total number of N patterns, the probability of correct classification can be calculated as:

$$P(c, N) = p_1 P(c, N \mid \varphi_1) + p_2 P(c, N \mid \varphi_2) \tag{3.26}$$

where $p_1 = p_2 = 1/2$ and $P(c, N \mid \varphi_1)$, $P(c, N \mid \varphi_2)$ as follows

$$P(c, N \mid \varphi_1) = \int_{\Omega_1(N)} f_{\mathbf{x}}(\mathbf{x} \mid \varphi_1) d\mathbf{x} \tag{3.27}$$

$$P(c, N \mid \varphi_2) = 1 - \int_{\Omega_2(N)} f_{\mathbf{x}}(\mathbf{x} \mid \varphi_2) d\mathbf{x} \tag{3.28}$$

and $\Omega_1(N)$ represents the region in the decision space where the multilayer perceptron classified the vector $\mathbf{x}$ as belonging to class φ_1. Example of how different layers of hidden neuron's affects the classification is illustrated in the graph below.

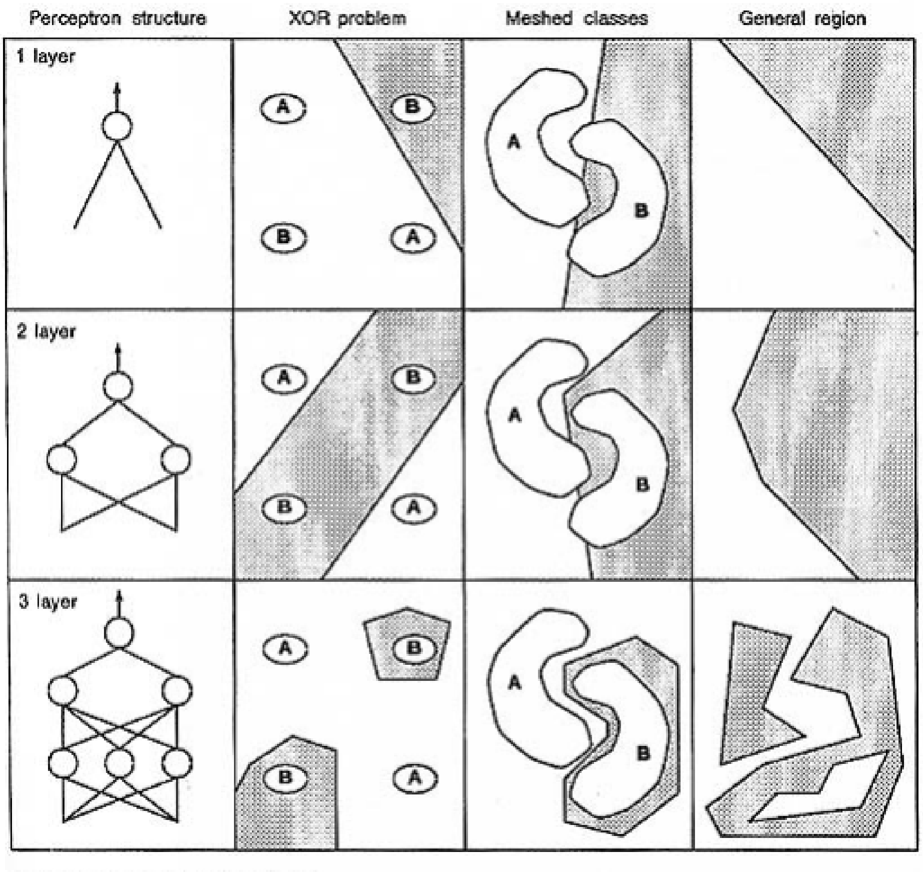


Figure 3.9: Illustraton of decision regions depending on layers.

This graph shows the decision regions (classification boundaries) that can be produced by different number of hidden layers. The network in the example is using the Heaviside function as a threshold function. For the MLP model the Signmoid function is recommended and would produce curved lines in the pattern space.

3.3 Training The Neural Networks

This section aims to present the theory of training the Neural Networks. There is a great variety of algorithms to train both deep and shallow networks, Back-Propagation Algorithm will be used to train the network in this thesis. The Back-Propagation Algorithm is often used to train perceptrons of different quantities of layers.

3.3.1 Back-propagation Algorithm

Back-propagation is a technique used for implementing *Gradient descent* for a multi-layer feedforward network. Where the basic idea is to compute partial derivatives of a function used for approximation such as $F(\mathbf{w}, \mathbf{x})$. The function $F(\mathbf{w}, \mathbf{x})$ is realized by the network with respect to all the elements of the weight vector $\mathbf{w}$ for a given value from input vector $\mathbf{x}$.

3.3.1.1 Steepest Descent

One of the most common methods for updating the weights of the neural network is the steepest descent method. The method updates the weights in the opposite direction to the gradient vector $\nabla\varepsilon(\mathbf{w})$. Where $\varepsilon = \frac{1}{2}e_j^2(n)$, and $e_j(n) = d_j(n) - h_j(n)$ representing the error term between the output of the network $h_j(n)$ and the desired response $d_j(n)$. Hence, the steepest descent method takes the form

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta\nabla\varepsilon(\mathbf{w}), \tag{3.29}$$

where η is the *learning rate.* From the step n to step $n+1$ the correction becomes

$$\Delta\mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n) = -\eta\nabla\varepsilon(\mathbf{w}). \tag{3.30}$$

Equation 4.49 can be used to approximate $\varepsilon(\mathbf{w}(n+1))$ using first order Taylor series expansion as follows

$$\varepsilon(\mathbf{w}(n+1)) \approx \varepsilon(\mathbf{w}(n)) + \nabla\varepsilon^T(n)\Delta\mathbf{w}(n). \tag{3.31}$$

Haykin (2009) shows that this rule fulfils the condition of iterative descent which corresponds to the following, consider a neuron j that receives the input signals $h_1(n), h_2, ...h_m(n)$ and responds to these signals by producing the output $v_j(n)$ where

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n)h_i(n). \tag{3.32}$$

As it was put forth earlier, to control for the bias in the model we think of $h_0 = 1$ corresponding to the bias with a weight $w_{j0} = b_j$. The output from the neuron passes through the activation function (see above) with another output as a result such as

$$h_j(n) = \varphi_j(v_j(n)). \tag{3.33}$$

The back-propagation algorithm applies a correction $\Delta w_{ji}(n)$ to the *synaptic weight*, the correction is proportional to the partial derivative $\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)}$. Using the chain rule of differentiation we can express the gradient as the following

$$\begin{aligned} \frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} &= \frac{\partial\varepsilon(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial h_j(n)}\frac{\partial h_j(n)}{\partial v_j(n)}\frac{\partial v_j(n)}{\partial w_{ji}(n)} \\ &= -e_j\varphi^{'}(v_j(n))h_i(n), \end{aligned} \tag{3.34}$$

where the derivatives of the error signal $e_j(n) = d_j(n) - h_j(n)$, the error energy $\varepsilon(n) = \frac{1}{2}e_j^2(n)$, the function signal $h_j(n)$ from the neuron j and local field $v_j(n)$ has been used. The partial derivative $\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)}$ above represents a *sensitivity factor* determining the direction of the search in weight space for the correct *synaptic weight* w_{ji}.

We use the delta rule for a correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ and is defined by

$$\Delta w_{ji}(n) = -\eta\frac{\partial\varepsilon(n)}{\partial w_{ji}(n)} \tag{3.35}$$

where η is the *learning-rate parameter* of the back-propagation algorithm. The use of the minus sign account for *gradientdescent* in weight space. In other words, it aims to reduce the value of $\varepsilon(n)$ when seeking for a direction o weight change. Replacing the result from the partial derivatives above in this formula yields

$$\Delta w_{ji}(n) = \eta\delta_j(n)y_i(n) \tag{3.36}$$

where $\delta_j(n)$ is the local gradient and is defined by

$$\begin{aligned} \delta_j(n) &= \frac{\partial\varepsilon(n)}{\partial v_j(n)} \\ &= \frac{\partial\varepsilon(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n)\varphi_j^{'}(v_j(n)) \end{aligned} \tag{3.37}$$

The local gradient described above, points at the required changes in synaptic weights in every iteration. According to the equations above, we can note that a key factor involved in calculating the weight adjustment $\Delta w_{ji}(n)$ is the error signal $e_j(n)$ at the output of the neuron j. Hence, there are two different cases to be considered, one where neuron j producing the output is in the output layer and one when it is in a hidden layer.

When neuron j is in the Output Layer

When neuron j is in the output layer, the error signal can be computed by using

$$e_j(n) = d_j(n) - y_j(n). \tag{3.38}$$

When the error signal is computed, it is fairly straight forward to compute the local gradient $\delta_j(n)$ using Equation 3.55 above.

When neuron j is a Hidden Layer

The local gradient can be re-written as the following when the neuron j is part of a hidden layer.

$$\begin{aligned} \delta_j(n) &= \frac{\partial \varepsilon(n)}{\partial h_j(n)} \frac{h_j(n)}{\partial v_j(n)} \\ &= \frac{\partial \varepsilon(n)}{\partial h_j(n)} \varphi_j^{'}(v_j(n)). \end{aligned} \tag{3.39}$$

To calculate the partial derivative $\frac{\partial \varepsilon(n)}{\partial y_j(n)}$ following Hayden (2009); if neuron k is an output neuron, the cost function is $\varepsilon(n) = \frac{1}{2}\sum_{k \in C} e_k(n)^2$. Making use of the cost function and putting it in the gradient of the cost function yields

$$\frac{\partial \varepsilon(n)}{\partial h_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial h_j(n)}. \tag{3.40}$$

As described above, the error is

$$e_k(n) = d_k(n) - h_k(n) = d_k(n) - \varphi_k(v_k(n)), \tag{3.41}$$

This gives

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k^{'}(v_k(n)). \tag{3.42}$$

Finally, using the features above, the gradient of the cost function becomes

$$\begin{aligned} \frac{\partial \varepsilon(n)}{\partial h_j(n)} &= -\sum_k e_k(n)\varphi_k^{'}(v_k(n))w_{kj}(n) \\ &= -\sum_k \delta_k(n) w_{kj}(n), \end{aligned} \tag{3.43}$$

where $v_k(n) = \sum_k w_{kj}(n)h_j(n)$ and $\frac{\partial v_k(n)}{\partial h_j(n} = w_{kj}(n)$. Finally, we obtain the back-propagation formula for the local gradient for hidden neuron j as the following

$$\delta_j(n) = \varphi_j^{'}(v_j(n)) \sum_k \delta_k(n) w_{kj}(n). \tag{3.44}$$

Short summary

To summarize the relations that has been derived for the black-propagation algorithm, we start with the correction $\Delta w_{ji}(n)$ applied to the synaptic weight that is connecting neuron i to neuron j, also called the delta rule:

$$\Delta w_{ji}(n) = \eta \delta_j(n) h_i(n) \tag{3.45}$$

Second, the local gradient $\delta_j(n)$ depends on whether the neuron j is in the output layer or hidden layer

1. If neuron j is in the output layer, $\delta_j(n)$ equals the product of the derivative $\varphi_j^{'}(v_j(n))$ and the error signal $e_j(n)$.

2. If neuron j is in the hidden layer, $\delta_j(n)$ equals the product of the associated derivative $\varphi_j^{i}(v_j(n))$ and the weighted sum of the δ's computed for the neurons in the next layers.

The Two Passes of Computation

In the application of the back-propagation algorithm, there is a distinction between two different passes of computation. Where the first pass is referred to as the forward pass and the second is the backward pass.

Forward Pass

In the forward pass, the input data passes through the synaptic weights from one layer of neurons to the next, until it finally passes through the output layer. The function signal is expressed as

$$h_j(n) = \varphi(\sum_{i=0}^{m} w_{ij}(n)h_i(n), \tag{3.46}$$

where φ represents the activation function. The total number of inputs passing through the network is w, applied to neuron j and w_{ji} is the synaptic weight connecting neuron i to neuron j.

Backward Pass

The backward pass, in contrast to the forward pass, starts at the output layer by passing the errors signals backwards through the network, layer by layer and computing the local gradients δ (see computations of δ for different activation function in Haykin (2009)) recursively for each neuron. The recursive process of this pass, permits the synaptic weights to undergo changes according to the delta rule explained above. This recursive process is continued layer by layer by propagating the changes to the synaptic weights.

4 Empirical Model

This chapter aims to present data used for the empirical models, the results from the empirical models and robustness checks. The main focus is the performance of the ANN model relative to other classification techniques. An attempt will be made to optimize the performance of the ANN model by finding the correct number of hidden layers that result in model highest classification accuracy. The data used for this study is in some manners unique, finding complete financial information on defaulted companies is not easy, for this reason, the meaning of "default" is used in a more broad manner, also including non-performing loans (NPLs). NPLs are loans for which corporates couldn't service the obligation to the debt-holders in 90-days or more. This information is still very relevant for the study because what is being investigated is the creditworthiness of companies.

4.1 Data

This section will be dedicated to present the data used for empirical application of the models. The differences in input variables used for credit scoring is highly diverse, this study aims to build on previous research but is restricted by data limitations.

4.1.1 Input Variable Selection

There are a large variety of financial items used in the literature for predicting creditworthiness. Traditional approaches on credit assessment such as Altmans Z-score and several other attempts described in the previous chapters have focused on quantitative input variables. Recent literature concludes that using only quantitative variables might not be sufficient for predicting defaults (Lehmann (2013), Grunet et al. (2004). Including qualitative variables such as legal form of business, number of employees, the region the business is operating on and industry type can increase the prediction power of the models. Noting but notwithstanding these findings, due to data-restrictions, this thesis will use quantitative items only.

There are several approaches to determine and select input variables for the models, Hand Henley (1997) puts forth three different approaches on this matter. First, and perhaps most important; expert knowledge can be used, this is more important when a model is being developed for a specific sector. Financial items can show great differences depending on which sector the company that is being assessed is operating on. Second, statistical procedures such as the forward and backward selection based on goodness of fit measures (R-squared) can be implemented. Third, to select variables by using a measure which indicates the difference between the distributions of the bankrupted and non-bankrupted companies on that variable, this can either be done by specific measures or simply illustratively with two-way plots of the variables. Also, other authors, such as Verstraeten and Van den Poel (2005) refer to the importance of the Receiver Operating Characteristic (ROC) Curve and its summary index Area Under the ROC Curve (AUC) in the explanatory variable selection process. The ROC Curve gives a graphical representation of the discriminatory power of a scoring system.

4.1.2 Input Variables

Financial standing of a corporate depends on many different factors, hence, insight from an experienced analyst is often hard to beat in the accuracy of judging the creditworthiness. It is however mentioned earlier that this approach is both time consuming and subject to human errors. This study aims to use quantitative data to model the creditworthiness. The concepts that will be looked at is the size of the company, profitability, leverage, liquidity and ability to cover interest costs. This concepts are widely used in corporate financial analysis.

Size
Size is related to volatility, which is inherently related to both the Merton and the Gambler's Ruin structural models. Smaller size implies less diversification and less depth in management, this implies greater susceptibility to idiosyncratic shocks. Size is also related to 'market position', a common qualitative term used when assessing creditworthiness.

Profitability
Profitability measures the degree to which a business is able to generate sales greater than the cost of operating. Companies must be profitable in the long-run, or at least generate cash flow to both survive and be creditworthy. Higher profitability should raise a firm's equity value. The measures also implies a longer way for revenues to fall or costs to rise before actual losses can occur.

Leverage
In addition to profitability, leverage is a key measure of credit risk. The higher the leverage, or gearing, the smaller the cushion for adverse shocks. The measures represent the difference between the funds supplied by the shareholders and the financing supplied by creditors. Higher leverage means the creditor is taking the risk as opposed to the shareholders.

Liquidity
Liquidity is a common variable in most credit assessments. The relevance for the credit assessment comes from the fact that liquidity is a necessary condition for

servicing debt.That is, if you have sufficient current assets, you can pay current liabilities. Liquidity is also a very powerful and obvious contemporaneous measure of default or creditworthiness, because if a firm is close to defaulting, its current ratio must be low.

Coverage

Coverage ratios reflects company's ability to pay the interest charges on its debt. The "coverage" aspect of the ratio indicates how many times the interest could be paid from generated earnings, and hence providing a sense of safety margin a company has for paying its interest for any period. A company that manages to generate earnings well above its interest payments,is in an excellent position to absorb possible financial storms and vice versa.

Concept	Financial Item	Calculation
Size	Total Assets	Value of Total Assets in 000' EUR
	Operating Revenue	Total Operating Revenue in 000' EUR
Profitability	Profit Margin	Net Operating Income/ Operating Revenue
Leverage	Gearing	Total Debt /Total Equity
	Solvency Ratio	Equity / Total Assets x 100
Liquidity	Current Ratio	Current assets / Current liabilities
Coverage Ratio	Interest Coverage	Operating income / Interest expense

Table 4.1: Input variables

4.1.3 Data Source

The firm-level data collected for this study is from the Bureau Van Dijk database Amadeus, containing comprehensive information on roughly 21 million companies across Europe. The data in the database is presented in a standardised way, for easier comparisons across countries and sectors. The database contains 1,8 million entities based in Sweden. The data quality and availability is however different across firms, for this reason, a comprehensive data quality ensuring process was followed. Sector specific models was decided against due to the restriction of available defaulted entities in many industries, hence a more general approach is followed. Having cleaned the dataset of missing values and randomised the selection of non-defaulted companies, the entire dataset for the modelling remains at 5926 observations where 5121 of them are still active and non-defaulted firms and 805 are companies that have defaulted on their payments.

4.1.4 Descriptive Statistics

The below tables presents descriptive statistics for the Swedish dataset, containing in total 5926 observations including 805 defaulted companies. Descriptive statistics is presented in three different versions, where the first graph shows descriptive statistics for the whole dataset, the second one on non-defaulted companies and the last table represents the defaulted companies.

Financial Items	Mean	Max	Min	Standard Deviation	Observations
Operating Revenue	99592,018	30133574,98	0,105	844063,575	5926
Toal Assets	110244,226	52851317,04	0,225	1234127,487	5926
Profit Margin(%)	5,467	100	-100	15,524	5926
Current Ratio	2,194	99,205	0	4,151	5926
Interest Coverage (x)	72,123	10000	-100	168,986	5926
Solvency ratio(x)	37,262	98,985	0,1	22,098	5926
Gearing (%)	102,600	997,794	0	175,548	5926

Table 4.2: Descriptive Statistics for the Swedish dataset.

The whole dataset contains 5926 observations, the observations for non-defaulted companies are randomly selected from a dataset representing both listed and non-listed companies in Sweden. Operating revenue and total assets are given in thousands of Euros, while other items are given either in percentage or multipliers.

Financial Items	Mean	Max	Min	Standard Deviation	Observations
Operating Revenue	115160,622	30133574,98	898,534	907012,694	5121
Toal Assets	127516,188	52851317,04	26,402	1326778,373	5121
Profit Margin(%)	5,927	99,692	-94,727	12,846	5121
Current Ratio	2,199	99,205	0	4,132	5121
Interest Coverage (x)	80,337	1000	-97,929	177,551	5121
Solvency ratio(x)	38,077	98,985	0,159	21,761	5121
Gearing (%)	101,443	994,973	0	170,857	5121

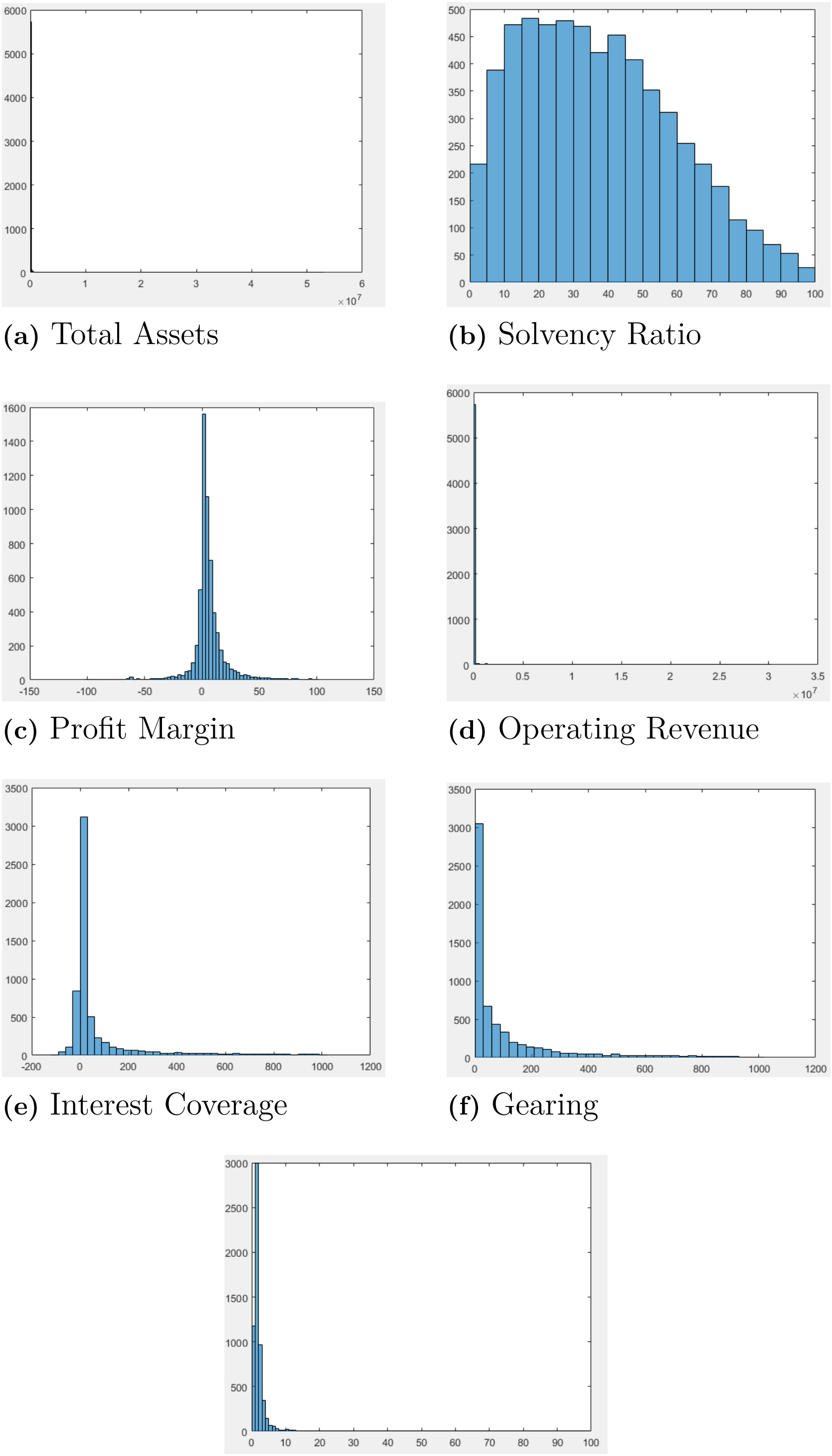
Table 4.3: Descriptive statistics for non-defaulted companies.

What is worth noting in the table above is that the mean value of operating revenue and total assets is higher than the mean in the dataset for all companies. This indicates that the companies in the non-defaulted dataset are systematically larger. The other major difference is in the interest coverage item, where the interest coverage among the non-defaulted companies are higher.

Financial Items	Mean	Max	Min	Standard Deviation	Observations
Operating Revenue	552,489	78110,407	0,105	3259,057	805
Toal Assets	368,795	30722,891	0,225	1697,099	805
Profit Margin(%)	2,546	100	-100	26,745	805
Current Ratio	2,156	78	0,004	4,277	805
Interest Coverage (x)	19,869	885	-100	80,788	805
Solvency ratio(x)	32,077	98,305	0,1	23,492	805
Gearing (%)	109,965	997,794	0	202,832	805

Table 4.4: Descriptive statistics for defaulted companies.

Again, what is of great importance to mention in the dataset for defaulted companies is the smaller size and difference in almost every other financial item. Just by looking at the descriptive statistics, one can note the clear stronger financial performance among non-defaulted firms. The distribution of all input variables are presented below.



(a) Total Assets

(b) Solvency Ratio

(c) Profit Margin

(d) Operating Revenue

(e) Interest Coverage

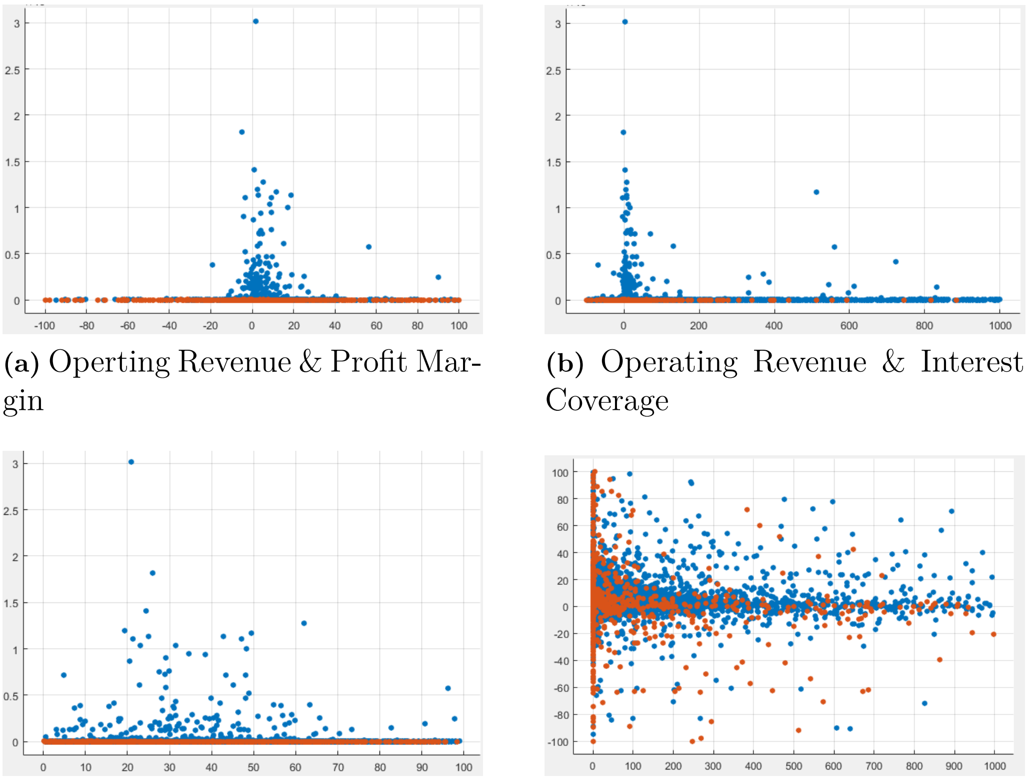
(f) Gearing

(g) Current Ratio

Figure 4.1: Histograms of input variables.

4.1.5 Feature Representation on a Two-dimensional Plane

Feature (input variables) selection constitutes one of the more important aspects of classification problems. One way of reaching an opinion on the separability of the classes "creditworthy" and "non-creditworthy" is to depict the features of companies in a two-dimensional space. Below, there are 9 graphs of two-way plots illustrated, the orange coloured dots represents the defaulted and hence non-creditworthy companies whereas the blue dots represents the non-defaulted and creditworthy companies. Looking at the graphs, it is quite clear that there are major areas with overlaps, this is not a surprise due to the fact that the selection of non-defaulted companies where randomly selected companies among all active companies. There is a great chance that the data of non-defaulted companies contains companies that are in financial distress but didn't yet default on any payments. In general what is really worth mentioning from the illustrations below is that, majority of the defaulted companies seems to be smaller in size. This attribute is measured in this study as operating revenue and total assets. Other important features in the separation of defaulted and non-defaulted companies are profitability, interest coverage and gearing ratios. Although there is a possibility to actually separate the companies by these features, one should highlight the existence of overlap in all features.



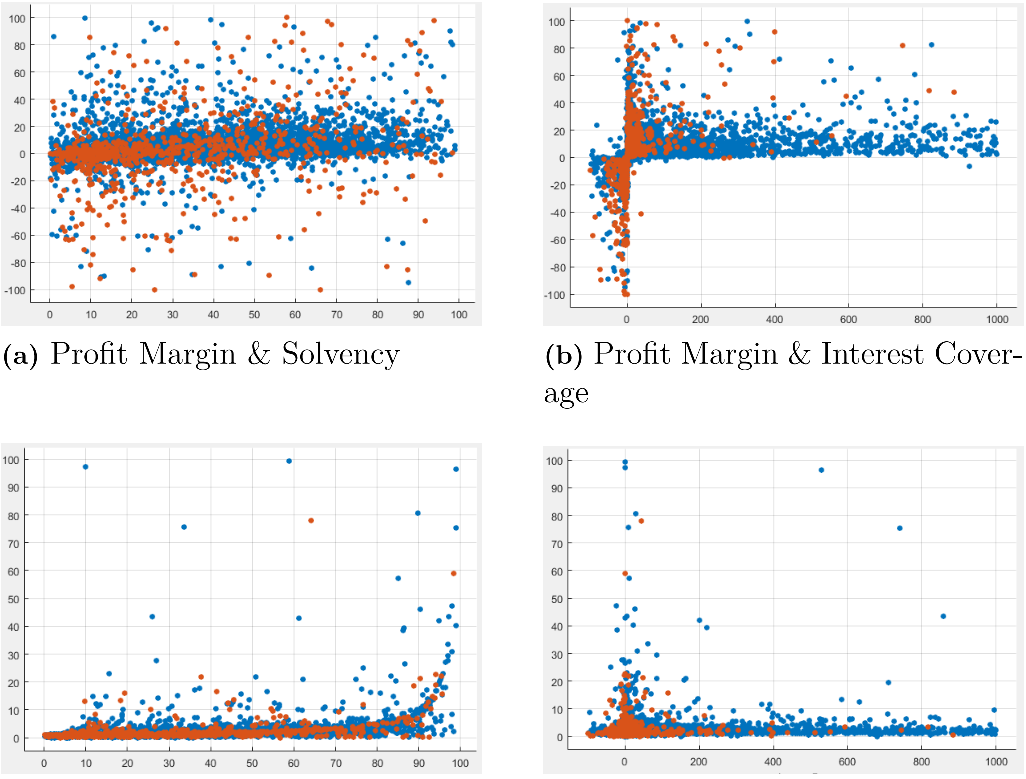
(a) Operting Revenue & Profit Margin

(b) Operating Revenue & Interest Coverage

(c) Operating Revenue & Solvenc Ratio

(d) Profit Margin & Gearing

Figure 4.2: Two-way plots of selected features 1.



(a) Profit Margin & Solvency

(b) Profit Margin & Interest Coverage

(c) Current Ratio & Solvency Ratio

(d) Current Ratio & Interest Coverage

Figure 4.3: Two-way plots of selected features 2.

4.2 Model

The empirical model to be applied to credit scoring of Swedish corporates will be as follows. The 7 input variables were described above. The benchmark ANN model will have 10 hidden layers but the effect of increase and decrease in hidden layers will also be discussed.

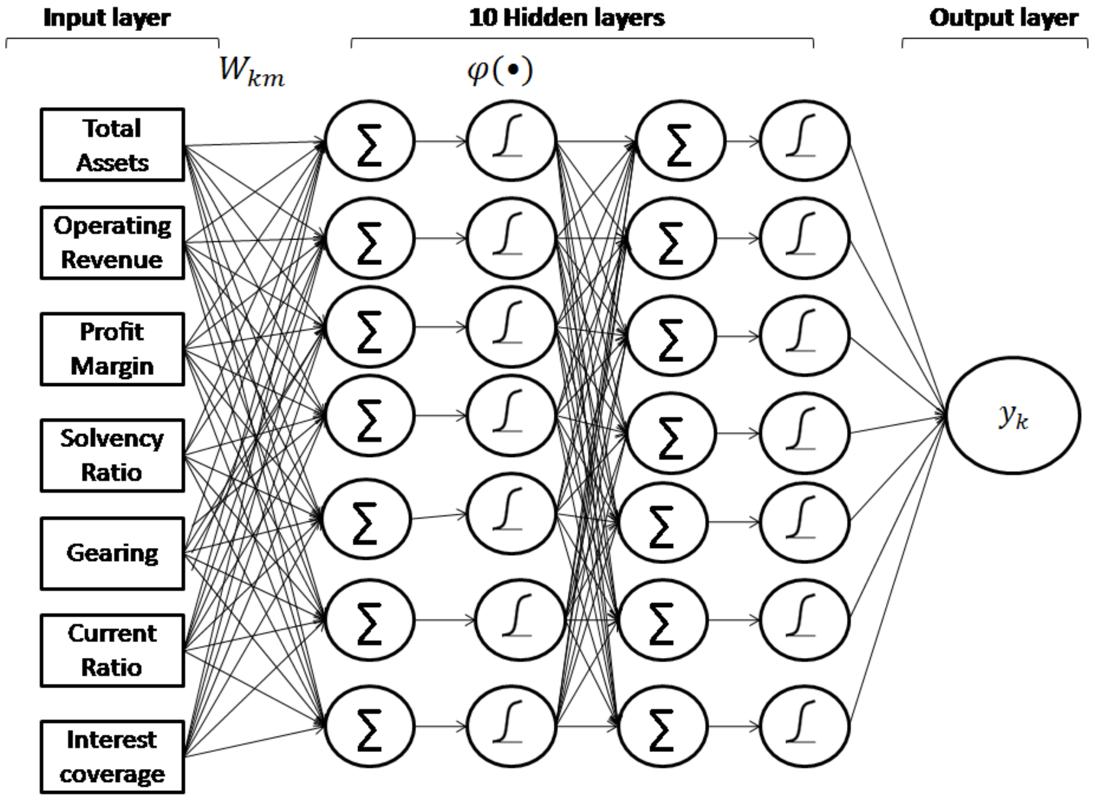


Figure 4.4: ANN 10-hidden layers benchmark model for Swedish non-financial corporates.

4.2.1 Training the Empirical Model

During the training, the weights and biases of the network were iteratively adjusted to minimize the network performance measure, e. The chosen measure represents the training error, calculated as the mean sum of squares of the difference between the output signal from the network, y_k, and the corresponding actual value, x_k. Error is measured between the actual and the desired outputs. This error is later back propagated and new weights are recalculated and thus neuron outputs are re-evaluated. The process is iterated until the error is minimized. This study have applied the following error function to optimize the model and prediction accuracy rates to evaluate their respective performance. The mean squared error function is defined as

$$e = \frac{1}{N}\sum_{k=1}^{N}(y_k - x_k)^2 \tag{4.1}$$

After each of the iterations during the network training, the ability of the network to be generalized was tested by performing simulations with the validation dataset. The agreement between the simulated y_k values, obtained for a given p_k vector of input variables, with the corresponding actual x_k values in the validation data was determined by the same error measure, e. When the e value increased or remained constant during 10 consecutive iterations, while the training error decreased, the network training was stopped.

4.2.2 Model Evaluation

A classification model can be deemed good in several ways, it can have a good predictive capability, scalability, how fast it takes to build the models and train them etc. This study will evaluate the models based on one criteria, the prediction accuracy, which is in line with studies such as Atiya, (2001), Gentry et al (1985), Etemadi (2009), Gurny Gurny, (2010). The prediction accuracy will be evaluated using confusion matrices and ROC curves. The ROC curve can be interpreted by looking at the position of the curve. The more the ROC curve is closer to y-axis, the better the estimated prediction model is. Or in other words, the model has higher discriminant power if its sensitivity and specificity are higher with respect to other model sensitivity and specificity. The curve finds its coordinates in the two-dimensional plane by comparing number of correctly and incorrectly classified classes to the total number of observations.

Accuracy in this study is defined in the same fashion as the previously mentioned similar comparative studies - the proportion of correct classifications. It is however worth noting that some errors can be more serious than others, and it may be important to control the error rate for some key classes. In general, classification problems have two types of errors, Type 1 (creditworthy is classified as non-creditworthy) and Type 2 (non-creditworthy is classified as creditworthy). In credit scoring it is believed that the cost of Type 1 and Type 2 errors are very different, it is substantially more costly to classify non-creditworthy as creditworthy than the other way around. Although this will be part of the analysis, it wont be a key determinant for the performance of the models in this study.

4.3 Result

This section will be devoted to show the result of the benchmark ANN model with 10 hidden layers, the effect of changes in hidden layers on classification accuracy, the result of applications of other non-parametric methods and robustness checks with data from Germany and France.

4.3.1 ANN Model Result

The ANN-model is applied to the Swedish data-set, where the training and testing of the model is done by separating the data as follows:

- 70% of the data for training
- 15% of the data for validation
- 15% of the data for testing

All samples are randomized from the whole dataset. The training sample is used to train the training and the network is adjusted according to error. The validation sample is used to measure network generalization and to halt the training when generalization stops improving. And the test sample is used to test the prediction accuracy of the network.

The following prediction accuracy is obtained from the benchmark model with 10 hidden layers.The ANN model with 10 hidden layers results in a prediction accuracy rate of 86.5%. Where most of the wrongly classified companies are defaulted companies.

	Creditworthy	Non-creditworthy
Creditworthy	5079 (85.6%)	706 (13.0%)
Non-creditworthy	42 (0.6%)	89 (0.8%)

Below is an illustration of Receiver Operating Characteristic (ROC) curve, one of the most common ways up illustrating the performance of a binary classifier. The blue line represent the models prediction accuracy, the closer it is to the diagonal line, the worse is the predictive power of the model. Looking at the ROC curve below, we can see that this is not really the case.

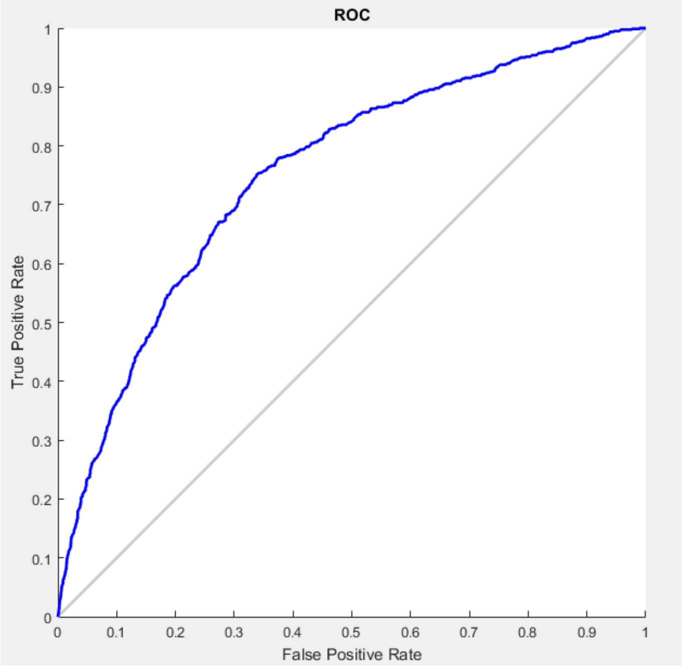


Figure 4.5: Receiver Operating Characteristic (ROC) curve for benchmark model

4.3.1.1 Different Hidden Layers

The effect of varying hidden layers has been showed to be critical in the studies of K.Chen (2012), I.Shafi (2006) and Karsoliya (2012). The benchmark model on Swedish data was initially trained with 10 hidden layers, the effect of variation in number of hidden layers on prediction accuracy is displayed below.

Number of Hidden Layers	Accuracy
5 Hidden Layers	85.41%
10 Hidden Layers	86.53%
20 Hidden Layers	86.57%
50 Hidden Layers	86.71%
100 Hidden Layers	87.49%
150 Hidden Layers	87.39%

Table 4.5: ANN Model with different hidden layers

Decreasing the number of hidden layers from 10 to 5 affects the prediction accuracy negatively, there was a decrease with almost 1%. Gradually increasing the hidden layers from 10 to 20 and later to 100 increased the performance by almost 1% compared to the benchmark model. The increasing effect is however not linear, trying 150 hidden had a marginally decreasing effect on prediction accuracy.

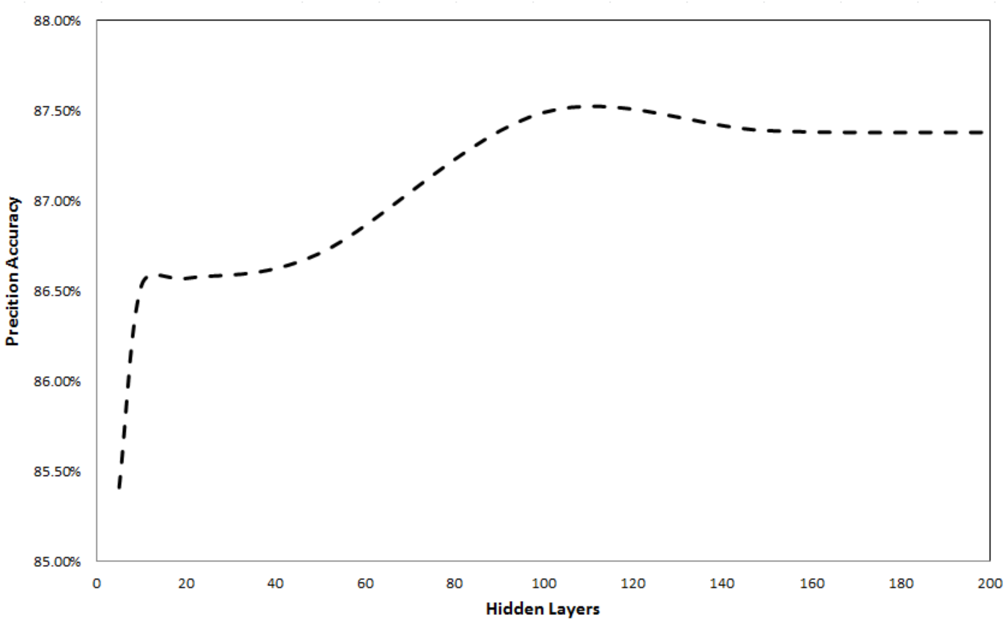


Figure 4.6: The effect of change in number of hidden layers on prediction accuracy.

4.3.2 Comparing the Methods

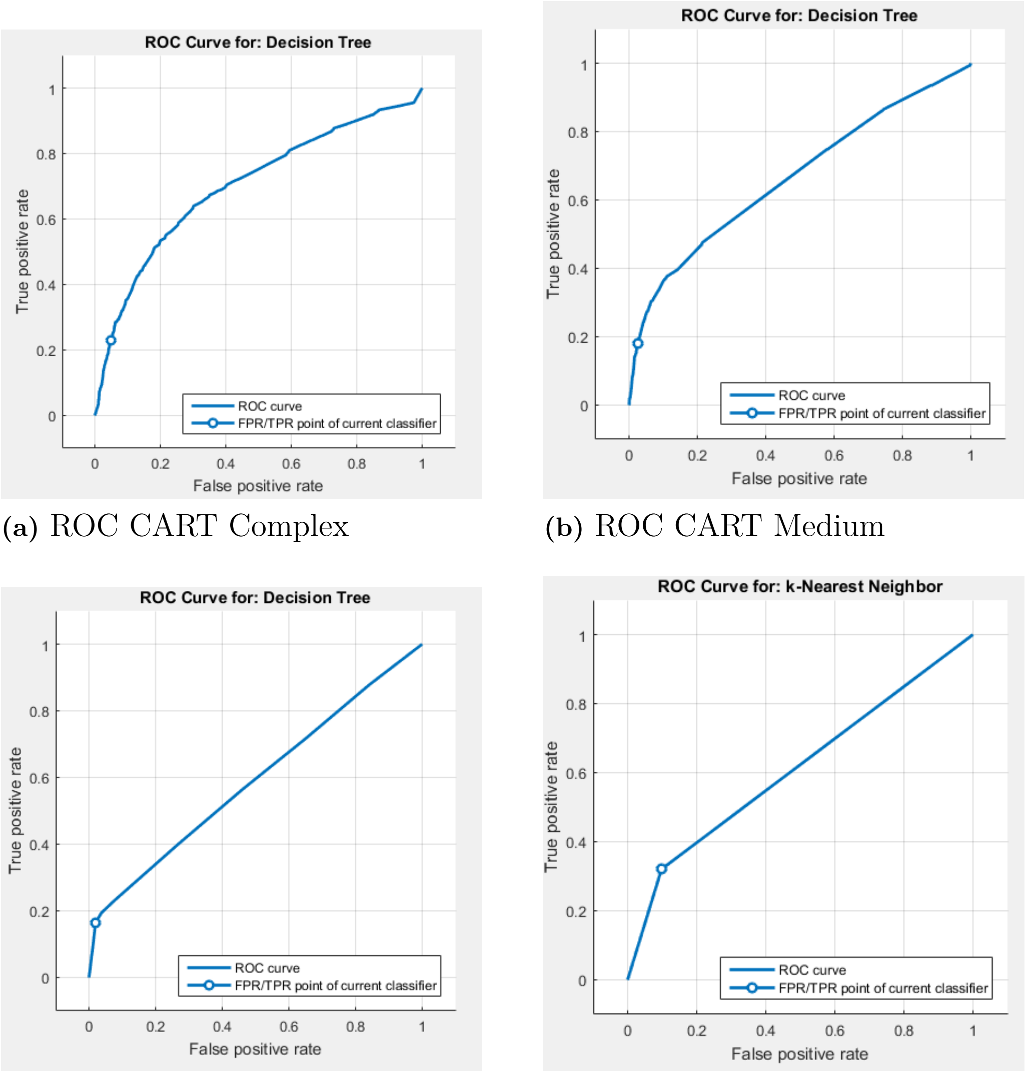
The performance of the optimized (with regards to hidden layers) ANN-model for Swedish corporates can only be understood when put in perspective and comparing it to other methods. The models that would be used for this purposed was described in the chapter for theoretical background. All of the techniques are characterized by their own strengths and weaknesses and they can perform differently with different datasets. The main ANN model with 150 hidden layers and the other 8 models were applied to the same dataset. The table below shows the prediction accuracy of the models. Again, 70% of the data was used for training the algorithms, 15% for validation and 15 % for testing.

Model	Accuracy
ANN 100 Hidden Layers	87.49%
CART Medium	86.53%
CART Complex	85.32%
CART Simple	86.90%
k-NN Fine	82.32%
k-NN Medium	84.36%
k-NN Coarse	83.57%
LDA	86.43%
QDA	72.31%

Table 4.6: ANN Model with different hidden layers

Although the difference is marginal, we can see from the table above that ANN outperforms the other models with a prediction accuracy of 87.49%. The second best performance is from the simple CART model with 86.9% prediction accuracy.

The worst performance is from the Quadratic Discriminant Analysis (QDA) model with 72.31% accuracy. The ROC curves for every model are shown in the graphs below.



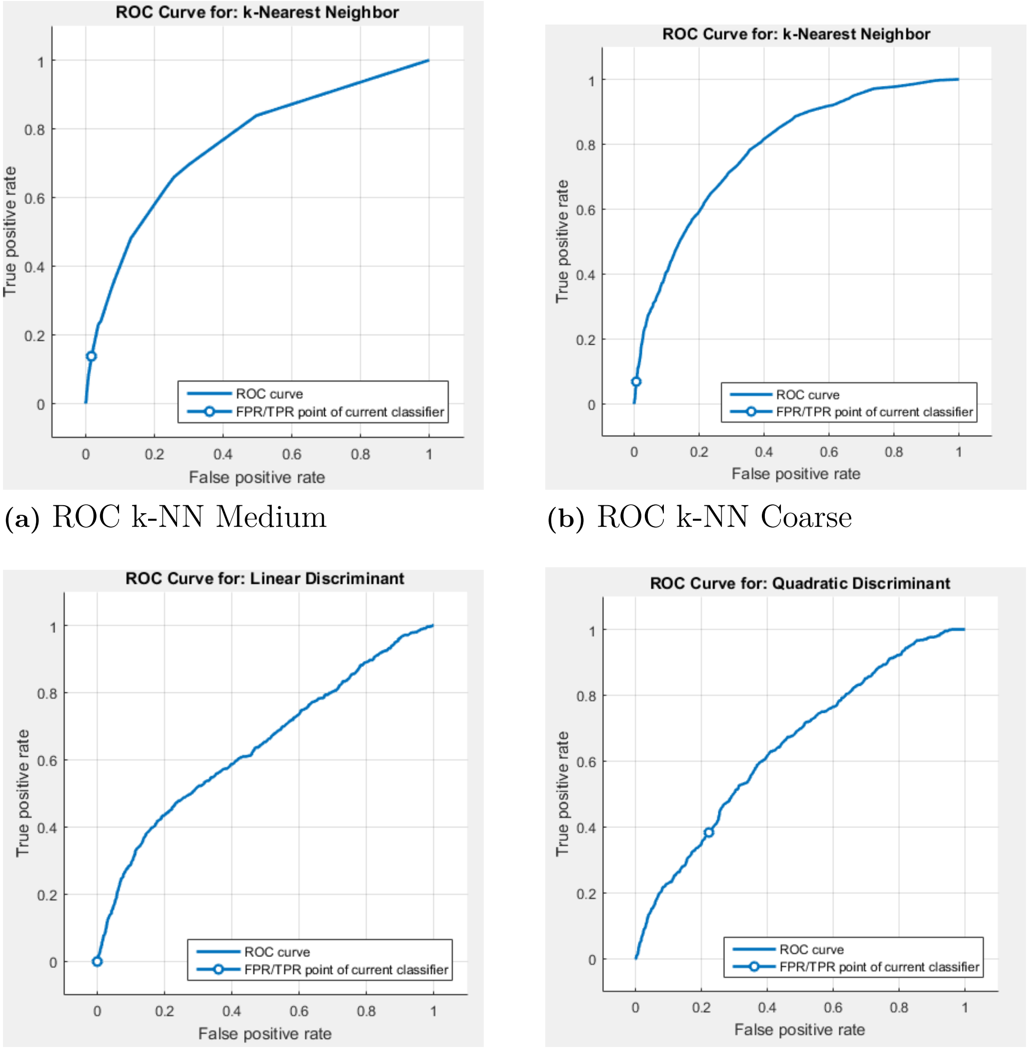
(a) ROC CART Complex

(b) ROC CART Medium

(c) ROC CART Simple

(d) ROC k-NN Fine

Figure 4.7: ROC curves for the models.



(a) ROC k-NN Medium

(b) ROC k-NN Coarse

(c) ROC LDA

(d) ROC QDA

Figure 4.8: ROC curves for the models.

4.4 Robustness Checks

One way to better understand and even validate the results presented above is to conduct "robustness checks". This section will be dedicated to apply the models to different datasets to look for inconsistencies or trends in the results. The results presented above are for the Swedish dataset which is the main focus of this study. The robustness checks starts with splitting the Swedish corporate dataset into three different groups and applying the same models to see if there are any changes in the results. The second and third part of the robustness checks consist of application of the models to two different datasets, namely French and German. There are no changes made on the models, variation is solely in data.

4.4.1 Splitting the Dataset

The rationale behind splitting the data comes from the differences that can be seen in the descriptive statistics section between non-defaulted and defaulted companies. The default among the companies with smaller size represented by the input variables total assets and operating revenues is substantially more common. For this reason, it is worthwhile to investigate whether the models perform differently among different portions of the dataset. Sorting the non-defaulted companies after largest by assets and revenue and later extracting top 25 %, 25-75% and bottom 25% gives three subsets. The models are applied to all of them.

4.4.1.1 Top 25%

The table below shows the prediction accuracies for the models that were applied to the dataset of 25% of the largest non-defaulted companies and the defaulted companies. It is worth mentioning the slight increase in performance in the top performing models ANN with 89,41%, simple CART 89,23%, CART medium 89,11% and complex CART 89,10%. The performance of k-NN and both forms of discriminant analysis decreased.

Model	Accuracy
Top 25%	
ANN	89.41%
CART Medium	89.11%
CART Complex	89.10%
CART Simple	89.23%
k-NN Fine	66.42%
k-NN Medium	72.93%
k-NN Coarse	70.00%
LDA	61.43%
QDA	81.32%

Table 4.7: Models prediction powers on Top 25 % by Operating Revenue.

4.4.1.2 25-75%

The second table illustrates the results for the largest corporates in the 25-75% segment. These corporates are still a bit larger than the average in defaulted companies but significantly smaller in size than the top 25%. The performance of the models dropped compared to the larger companies but it is still slightly higher than the application of the models on the original dataset. Best performing model in this segment is again ANN (88.3%) followed by the CART models .

Model	Accuracy
25-75%	
ANN	88.3%
CART Medium	86.63%
CART Complex	86.20%
CART Simple	87.41%
k-NN Fine	81.30%
k-NN Medium	81.80%
k-NN Coarse	74.93%
LDA	71.61%
QDA	75.21%

Table 4.8: Models prediction powers on 25-75 % by operating revenue.

4.4.1.3 Bottom 25%

The third table represents the dataset with the smallest 25% non-defaulted corporates. The performance of every model drops significantly with none of them being able to reach a prediction accuracy of 80%. The performance is both lower compared to the segments presented above and the original dataset. ANN seems to be the best performing model once again, followed by CART and k-NN.

Model	Accuracy
Bottom 25%	
ANN	79.24%
CART Medium	65.74%
CART Complex	67.21%
CART Simple	66.16%
k-NN Fine	65.00%
k-NN Medium	68.42%
k-NN Coarse	66.12%
LDA	63.00%
QDA	62.6%

Table 4.9: Models prediction powers on bottom 25 % by operating revenue.

4.4.2 Other Datasets

The second part of the "robustness checks" is to apply the models on other datasets. The original dataset comprising Swedish corporates consisted of roughly 14% defaulted companies. To be able to compare the results in a fair way and ensure valid results, the same proportion of defaulted companies were kept in the German and French datasets.

The choice of other dataset was based on the similarities between the countries, optimally; dataset from other Scandinavian would have been used. However, the lack of data on defaulted companies in that region prohibited this option. Both France and German are two EU-member countries with similar ecosystems for corporates, even if the choice is not optimal, for comparison reasons these datasets are similar in character.

4.4.2.1 Germany

The German dataset consists of 1511 observations where 13.9% of those are defaulted corporates from different industries. The table below shows the descriptive statistics for the dataset.

Financial Items	Mean	Max	Min	Standard Deviation	Observations
Operating Revenue	1981265,045	212756000	7,14	266515,534	1511
Toal Assets	12990196,312	351210000	21,481	364224,717	1511
Profit Margin(%)	4,233	98,43	-99,783	14,487	1511
Current Ratio	3,483	93,307	0,001	7,447	1511
Interest Coverage (x)	23,506	996,089	-92,927	79,232	1511
Solvency ratio(x)	37,980	99,516	0,218	21,705	1511
Gearing (%)	12,952	997,964	0	171,169	1511

Table 4.10: Descriptive Statistics for full dataset Germany

The German dataset compared to the Swedish dataset contains a lot larger companies with mean of 1,981,261 thousand euros in operating revenue. There are also minor differences in profitability, capital structure and interest coverage capabilities.

The prediction accuracies from the models applied on German dataset is shown in the table below. ANN is again the best performing model with 87.54% accuracy, it is followed by k-NN and QDA seem to be the worst performing model. The result from the German dataset is fairly similar to the results from the original dataset.

Model	Accuracy
Germany	
ANN	87.54%
CART Medium	85.34%
CART Complex	82.81%
CART Simple	86.26%
k-NN Fine	81.60%
k-NN Medium	87.42%
k-NN Coarse	87.31%
LDA	86.00%
QDA	28.81%

Table 4.11: Models prediction accuracy for German dataset.

4.4.2.2 France

The largest dataset used in this study is the French dataset of non-financial corporates. The dataset contains 25832 observations with roughly 14% of them being defaulted companies. The descriptive statistics for the dataset is presented in the table below.

Financial Items	Mean	Max	Min	Standard Deviation	Observations
Operating Revenue	122502,9	174857009	9,078	1807354,611	25832
Toal Assets	144717,2	26798900	8,805	2850553,434	25832
Profit Margin(%)	3,324	99,419	-99,542	0,058399	25832
Current Ratio	1,608	96,8	0,001	2,167	25832
Interest Coverage (x)	45,763	999,468	-99,876	121,253	25832
Solvency ratio(x)	34,087	99,959	0,003	19,647	25832
Gearing (%)	97,153	999,738	0	149,531	25832

Table 4.12: Descriptive Statistics for full dataset France.

The French dataset, as it was with the German dataset contains on average larger companies with 1,225,02 thousand euros in operating revenue. It is slightly smaller than the average in the German dataset. There are major differences in the other input variables compared to both the Swedish and German dataset.

The prediction accuracies from the models applied on French dataset is shown in the table below. These results are very different from the previously obtained results. As it can be seen, the prediction accuracy is for the first time above 90% and the best performing model is the medium k-NN with 98.52% prediction accuracy. ANN performs better than it did on the Swedish and German dataset with 91.14%.

Model	Accuracy
France	
ANN	91.14%
CART Medium	98.42%
CART Complex	98.33%
CART Simple	98.26%
k-NN Fine	97.31%
k-NN Medium	98.52%
k-NN Coarse	98.43%
LDA	85.30%
QDA	35.24%

Table 4.13: Models prediction accuracy for French dataset.

5
Discussion

The objective pursued by this study is to model Swedish corporate creditworthiness with Artificial Neural Networks (ANN) and compare the results to other commonly used parametric and non-parametric methods such as CART, k-NN, LDA and QDA. In the above chapters, an introduction to credit assessment, theoretical background for ANN and other techniques were presented. The chapter dedicated to show the empirical investigations lays the ground for this chapter. With the light shed from previous studies and theoretical considerations, the results of this study will be discussed.

5.1 Performance evaluation of ANN model

The ANN-models have been shown to outperform many conventional methods in credit scoring such as regression-models, discriminant analysis, CART, k-NN and several others (Rafiei et al. (2011) Etemadi et al. (2009)). Previous studies such as Atiya (2001) and M. Al Doori & B.Beyrouti (2014) finds prediction accuracies for credit scoring with ANN's of 85.5% and 87.4% respectively. The result from this paper is in line with similar studies for both the level of accuracies and the performance of the ANN compared to other models.

The initial model with 10 hidden layers had a performance of 86.5%, one of the aims with this study is to look at how the variation in number of hidden layers affects the performance of the model. Determining the optimal amount of hidden layers is a critical issue for the model performance. If it is too small, the network cannot process sufficient information (even more important with complex models with many input-variables), and thus result in an inaccurate classification performance. And if the number of hidden layers are too many, the training process will be very long and require more number of epochs to end the training (D. Srinivasan, 1994). Up to the knowledge of some researchers, there is no absolute criteria to determine the exact number of hidden neurons that might lead to an optimal solution. Different number of hidden neurons are used in; Arai (1993), Atiya, (2001), M. Al Doori & B.Beyrouti, (2014) and D. Srinivasan (1994) argues that the appropriate number of hidden layers is system dependent, mainly determined by the size of the training set and the number of input variables. For this reason, different number of hidden layers was tested; the simulations resulted in the fact that the performance decreased when the hidden layers decreased from 10 to 5 and increased with every increasing change until 100 hidden layers. By trying different types of hidden layers, the performance of the model was increased from 86,53% to 87,44%, almost 1% increase.

From the descriptive statistics, especially two-way plots, it can be seen that there are a lot of overlaps between the defaulted and non-defaulted companies. This is not very surprising, because there are companies that are financially weak but didn't yet default on any payments. One could argue that the purpose of credit assessment-models is to be able to separate out precisely the companies that are almost about to default from the ones that actually defaults. This is however easier said than done, it could even be very difficult with insight from an experienced analyst hence mathematical models should be evaluated with this difficulty in mind. There are no magical input variables that easily and accurately separates the firms. A firm can be small but have very sound finances; it could be large with weak financials but still manages to keep the operation running. It is clear from the confusion matrices that, the model struggles with the over-lapping part of the data, hence most of the errors are made in the prediction of defaulted and not non-defaulted.

This study was conducted with relatively small number of input variables, although some older models look at 4-5 variables only, there are studies were a lot more variables are used (M. Al Doori& B.Beyrouti, 2014). The aim was to look at the major determinants of a company's financials namely; size, profitability, leverage, liquidity and coverage. These determinants can be measured in many different ways and some of them are highly correlated. The choice of input variables was based on previous studies on credit assessments and the availability of data. It's important to emphasize the importance of good quality data, if the model is to be applied on various companies in different sectors, it is important to choose input variables that are available for all companies, no matter industry. The ANN-model offers further topics to investigate, the accuracy of the model can possibly be improved by building sector-specific models, although there are challenges to get enough data on defaulted companies in some sectors. Further, this study does not take into account any macroeconomic aspects where for example the rise in interest rates could be crucial for the interest expenditure of some companies and this could also improve the accuracy of the model. Another aspect that was not taken into account in this study is the qualitative characteristics of the companies, such as management quality. This could be incorporated into the model by quantifying the quality of management by looking at the number of years of experience in the management of the companies but could require substantially more work.

5.2 Performance evaluation of ANN compared to other models

Evaluating performance of the ANN-model by simply looking at the prediction accuracy of that model does not really give a fair view of the quality of the model. This is the main reason other models were added to this study. The models used are commonly used to compare the performance of cutting-edge techniques such as ANN to more established models. The most successful models in terms of prediction accuracy is the ANN with 87.49 % prediction accuracy. Given that the ANNs allow for more non-linearity in the relationship, this result is perhaps not that surprising. If we look at the graphs showing how the defaulted and non-defaulted companies are distributed in a two-way plot we can see that it is hard to separate those classes linearly. The results of this study are consistent with other studies when credit scoring data is used, but inconsistent with some others where the models are applied to other type of datasets. For example in a three way classification, Desai etal. (1997) found that when considering the predictive performance, LDA was almost identical to ANN and slightly better than CART. But ANN gave the greatest predictive performance when predicting default (rather than non-default). King et al. (1994) found that LDA shows a greater predictive performance than neural networks, but a poorer performance compared with CART. On the other hand, Desai et al. (1996) in a two way classification found that LDA was inferior to ANN at predicting both defaulted and non-defaulted.In addition Khoylou & Stirling (1993) using a sample of two thousand cases,found that neural networks performed considerably better than multiple linear regression.

The results from the above mentioned articles are in line with the result in this study; ANN outperformed the other models followed by LDA, CART and k-NN. However, as there is no universally best method that works best for all kind of datasets, the performance is almost purely data driven. Because of the complexity and non-linearity of credit assessment models, the models that are more flexible in carving out the classes from each other performs the best. ANN in comparison to other methods is flexible and flexibility can be increased by increasing hidden layers (see optimal number of hidden layers section above). Because the performance of the model is so much dependent on the data, it is of crucial importance to try to apply the models on different datasets. For this reason, the models were applied in both different sections within the same dataset (Swedish dataset) and to datasets from other countries with different sizes.

5.3 Robustness

As it was mentioned above, the evaluation of the performance of a model cannot be done by simply looking at the prediction accuracy of one model. The performance rather is looked at in comparison to the performance of other models and consistency across datasets. The original dataset of Swedish corporates was split into three parts after the size of the companies measured by total operating revenue. It was mentioned above that the models struggles to separate the companies that are in overlap with the defaulted one, this investigation is an attempt to show that this is the case.

The first part of this analysis is to apply the model on the dataset with the largest non-defaulted corporates and defaulted corporates, almost all of the models performs better at predicting default among these companies. With ANN at 89.41% predicting accuracy followed by CART around 89%. Looking at how the models perform on the mid-sized companies in the dataset we can see that the prediction accuracy drops to 88.3% for ANN and 86% for CART which indicates that there is a larger overlap among these firms. What is of great importance is the performance of the models for the bottom 25% sized companies. There is a substantial decrease where ANN shows a prediction accuracy of 79.24% and the other models fall below 70%. These results shows in a more concrete way precisely where the models struggle to separate the defaulted companies from the non-defaulted. The fact that ANN performs best among the models is consistent with the findings of Desai et. al. (1997) where the ability of ANN to separate the defaulted is highlighted.

Another important aspect of the robustness checks is the application of the models to different datasets. Datasets with corporate level data from Germany and France was used for this purpose. There were no changes made on the models specifications. The German dataset in comparison to the Swedish and French dataset contains less observations with 1511 companies, however the proportion of defaulted companies within this dataset is similar to the other datasets of roughly 14%. On average, the companies in the German dataset are larger compared to the other datasets. ANN shows once again the best prediction accuracy among the models with 87,54% which is slightly larger than the Swedish dataset. This increase could be due to the existence of larger companies in the German dataset (the top 25% largest in the Swedish dataset was also easier to separate). The result from the original dataset is consistent over the German dataset where the ANN outperforms the other models.

The French dataset in comparison to the Swedish and the German one is very different in size, it contains 25,832 observations. In the study of Khoylou Stirling (1993), the size of the sample in which the models are training and tested on affects the prediction accuracy of the models. A similar pattern can be seen in this study, where the prediction accuracy of almost every model increased to above 90%. Similar findings are presented in Rui A.A. El-Keib (2002), where smaller sample sizes affects the prediction performance of some models more than others. In the case of France, ANN is no longer the best performing model k-NN shows a prediction accuracy of 98.52% compared to the 91.14% of ANN. It is hard to find a convincing argument for why the performance of these models changed in this direction when the dataset changed. However, it was mentioned earlier that the performance of many machine learning algorithms, including the ANN is dependent on the data. This change in performance compared to the Swedish and German dataset emphasizes that finding. One should keep in mind that although the non-defaulted companies in the datasets were randomized, it is likely that there are some kind of bias in the datasets can be responsible for the changes in prediction accuracies. Another possible explanation for the differences in the performance between different datasets is the macroeconomic conditions, which this studies does not consider. It could be the case that some macroeconomic condition in a country leads to more defaulted companies and even more easily separable companies within that country.

6

Conclusion

This study reviews the problem of bankruptcy prediction (creditworthiness) with artificial neural networks (ANN). Previous studies suggest that ANNs outperform other statistical methods such as regressions, discriminant analysis (LDA and QDA) and also non-parametric methods such as k-NN and CART. This study is among the first attempts to apply ANNs on Swedish corporate level data. The findings suggest that the ANN model outperforms the other models with a prediction accuracy of 87,44%. It is closely followed by CART and QDA is the worst performing model. The effect of number of hidden layers in the networks are also investigated and as it is in previous studies, the increase in number of hidden layers improved the performance of the 86,53%(10 hidden layers) to 87,49%(with 100 hidden layers). The increase is however not linear and it starts to decrease shortly after 100 hidden layers.

Number of Hidden Layers	Accuracy
5 Hidden Layers	85.41%
10 Hidden Layers	86.53%
20 Hidden Layers	86.57%
50 Hidden Layers	86,71%
100 Hidden Layers	87.49%
150 Hidden Layers	87.39%

Table 6.1: ANN Model with different hidden layers

The ANN model, together with the other models was applied to both different sections within the original dataset and to other datasets. The results suggest that the model struggles to separate the overlapping of non-defaulted and defaulted companies that is happening among the smallest firms in the non-defaulted dataset and the defaulted dataset. The prediction accuracy of the models are consistent over the German dataset where ANN outperforms the other models however the application on the French dataset with substantially more observations resulted in a higher prediction accuracy for all models and both k-NN and CART outperformed ANN.

Model	Sweden	Germany	France
ANN	87.49%	87.54%	91.14%
CART Medium	86.53%	85.34%	98.42%
CART Complex	85.32%	82.81%	98.33%
CART Simple	86.91%	86.26%	98.26%
k-NN Fine	82.32%	81.60%	97.31%
k-NN Medium	84.36%	87.42%	98.52%
k-NN Coarse	83.57%	87,31%	98.43%
LDA	86.43%	86.00%	85.30%
QDA	72.31%	28.81%	35.24%

Table 6.2: Models prediction accuracy for different datasets

6.1 Limitations

It is mentioned in the discussion chapter that the model can only be as good as the input data, the lack of data on defaulted companies puts limitations on the size of the dataset which can affect the results. Furthermore, this study evaluates the model performances with one criteria; the overall classification accuracy, and does not take into account the Type-2 errors that can be of great importance for a lender. A model that show higher overall accuracy does not necessarily have less Type-2 errors, hence the performance measure could be viewed critically. Another leap this study makes is the classification of "non-creditworthy" if the company defaults (according to regulation in that country) or doesn’t service debt within 90-days or longer, obviously the regulation for bankruptcy can be different among countries and a "defaulted" company in Sweden might legally survive in France due to other criteria.

6.2 For Further Studies

This study puts forth an empirical evaluation of two parametric (LDA and QDA) and three non-parametric methods (k-NN, CART and ANN) for the classification of creditworthy and non-creditworthy companies. Although it performs some optimization efforts for CART, k-NN and ANN, the models are kept in relatively primitive versions. One way to take this study further would be to enhance the performance of all of the models by for example pruning the CART-model (P. Tomayo, 2000), finding optimal k in k-NN (Hassanat, 2014) or choosing optimal error functions in ANN (Gangal, 2007). After these optimizations, the models could be tested again.

Another way would be to use different performance measures, this study looks at prediction accuracy by overall correctly classified corporates, however, the importance of Type 2 errors in credit scoring is noted. A comparative study on the Type 2 errors of similar models could be of great value for both practical and academic reasons.

Bibliography

[1] Abdou, H., Pointon, J. (2009). Credit scoring and decision-making in Egyptian public sector banks. International Journal of Managerial Finance 5(4), 391-406.

[2] Altman E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23 (4), 589-609.

[3] Amir Atiya. (2001) Bankruptcy prediction for credit risk using neural networks: a survey and new results. IEEE Transactions on Neural Networks, Vol. 12, No. 4, 929-935.

[4] Aziz, A., Davit, C.E., Gerald, H.L. (1988). Bankruptcy prediction – an investigation at cash flow based models. Journal of Management 25, 419–437.

[5] Bailly, J. S., Arnaud, M., Puech, C. (2007). Boosting: A classification method for remote sensing. International Journal of Remote Sensing, 7, 1687-1710.

[6] Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6), 627–635.

[7] Black, Fischer and Myron S. Scholes, (1973). The Pricing of options and corporate liabilities, "Journal of Political Economy 81, 637-654.

[8] Boyes, William J., Dennis L. Hoffman and Stuart A. Low, (1989), An Econometric Analysis of the Bank Credit Scoring Problem, Journal of Econometrics 40, 3-14.

[9] Buhlmann, P., Yu, B. (2003). Boosting with the L2 loss: Regression and classification. Ameri-can Statistical Association, 98, 324-339.

[10] Coffman, J. Y. (1986). The proper role of tree analysis in forecasting the risk behaviour of borrowers, Management Decision Systems, Atlanta, MDS Reports 3,4,7, and 9.

[11] Devaney, S., (1994). The Usefulness of Financial Ratios as Predictors of Household Insolvency: Two Perspectives. Financial Counseling and Planing,

vol. 5, 15-24.

[12] Dudoit, S., Fridlyand, D., Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association, 97, 77-87.

[13] Durand, D. (1941). Risk Elements in Consumer Instalment Financing, Studies in Consumer Instalment Financing. New York: National Bureau of Economic Research.

[14] T. C. Fogarty N. S. Ireson. (1993). Evolving Baysian classifiers for credit control—a comparison with other machine learning methods. IMA J. Math Appl. Business and Industry 5,63-75.

[15] Gentry, J.A., David, T.W., Paul, N., (1985). Classifying bankrupt firms with fund flow components. Journal of Accounting Research 23 146–160.

[16] Foreman, R.D., (2003). A logistic analysis of bankruptcy within the US local telecommunications industry. Journal of Economics Business 55, 135–166

[17] F.M. Rafiei, S.M. Manzari, S. Bostanian,(2011). Financial health prediction models using artificial neural networks, genetic algorithm and multivariate discriminant analysis: iranian evidence, Expert Syst. Appl. 38, 10210–10217.

[18] Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7 (2): 179-188.

[19] Frydman, H. – Altman, E. – KAO, D.-L. (1985). Introducing Recursive Partitioning for Financial Classification: The Case of Financial Distress. The Journal of Finance, vol. 40, 269-291.

[20] Gurný, P., Gurný, M. (2010), Comparison of the Credit Scoring Models on PD Estimation of US Banks. Mathematical Methods in Economics.

[21] Gurný, P., Gurný, M. (2009), Estimation of PD of Financial Institutions within Linear Discriminant Analysis. Mathematical Methods in Economics.

[22] Guillen, M., Artis, M. (1992). Count Data Models for a Credit Scoring System: The European Conference Series in Quantitative Economics and Econometrics on Econometrics of Duration, Count and Transition Models. Paris.

[23] Hart, A. (1992). Using neural networks for classification tasks: Some experiments on datasets and practical advice. Journal of the Operational Research Society, 43, 215-226.

[24] Harrell, F.E., Lee, K.L., (1985). A comparison of the discrimination of discriminant analysis and logistic regression. In: Se, P.K. (Ed.), Biostatistics: Statistics in Biomedical, Public Health, and Environmental Sciences. North-Holland, Amsterdam.

[25] H. Etemadi, A. Rostamy, H. Dehkordi,(2009). A genetic programming model for bankruptcy prediction: empirical evidence from Iran, Expert Syst. Appl. 36 (2) 3199–3207.

[26] Huang, C.-L., Chen, M.-C., and Wang, C.-J.(2007). Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications 33, 4 , 847-856.

[27] I. Shafi, J. Ahmad, S. I. Shah and F. M. Kashif.(2006). Impact of Varying Neurons and Hidden Layers in Neural Network Architecture for a Time Frequency Application Multitopic Conference, 2006. INMIC '06. IEEE, Islamabad, 2006, pp. 188-193.

[28] Jarrow, Robert ,(2001), "Default Parameter Estimation Using Market Prices" Financial Analysts Journal 57, pp. 75-92.

[29] J. Khoylu M. Stirling (1993). Credit Scoring and Neural Networks, Presented at Credit Scoring and Credit Control conference, Edinburgh.

[30] Jones, P., Mason, S., and Rosenfeld, E., (1984), “Contingent Claim Analysis of Corporate Capital Structures: An Empirical Investigation,” Journal of Finance 39, 611-625.

[31] Joachims, T. (1998). Text categorization with support vector machines. In Proceedings of European conference on machine learning (ECML), Chemintz, DE, pp.137–142.

[32] K. Chen, S. Yang and C. Batur. (2012). Effect of multi-hidden-layer structure on performance of BP neural network: Probe. Natural Computation (ICNC), 2012 Eighth International Conference on, Chongqing, , pp. 1-5.

[33] Kuhnert, P. M., Mengersen, K., Tesar, P. (2003). Bridging the gap between different statisti-cal approaches: An integrated framework for modeling. International Statistical Review, 71, 335-368.

[34] Liu, D., Chun, Y, (2009). The effects of different classification models on error propagation in land cover change detection. International Journal of Remote Sensing, 20, 5345-5364.

[35] Li, J., Liu, J., Xu, W., and Shi, Y. (2004). Support vector machines approach to credit assessment. In Computational Science-ICCS 2004. Springer, pp. 892-899.

[36] Makowski, P. (1985). Credit scoring branches out. The credit management: a survey. Operations Research 42, Credit World 75, 30–37.

[37] Manuel A. Hernandez • Maximo Torero (2014). Parametric versus nonparametric methods in risk scoring: an application to microcredit. Empirical Econonmic, 46:1057–1079.

[38] Martin, D., (1977). Early warning of bank failure. Journal of Banking and Finance 1, 249–276.

[39] Merton R. (1974). On the pricing of corporate debt: The risk structure of interest rates. The Journal of Finance, 29 (2), 449-470.

[40] Meyer, P.A., Pifer, H.W., (1970). Prediction of bank failures. Journal of Finance 25, 853–858.

[41] Myers, J. H., Forgy, E. W. (1963). The development of numerical credit evaluation systems. Journal of Ameri- can Statistics Association 58(September), 799–806.

[42] Min, S. H., Lee, J., Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert System with Applications, 31, 652–660.

[43] Preatoni, D. G., Nodari, M., Chirchella, R., Tosi, G., Wauters, L. A., Martinoli, A. (2005). Identifying bats from time-expanded recordings of search calls: Comparing classification methods. Journal of Wildlife Management, 69, 1601-1614.

[44] R. D. King, R. Henry,C. Feng A. Sutherland.(1994). A comparative study of classification algorithms: statistical, machine learning and neural network. Machine Intelligence, vol 13, (K. Furukwa, D. Michie S. Muggletoneds). Oxford: Oxford University Press.

[45] Reibnegger, G., Weiss, G., Werner-Felmayer, G., Judmaier, G., Wachter, H. (1991). Neural networks as a tool for utilizing laboratory information: Comparison with linear discriminant analysis and with classification and regression trees. Proceedings of the National Academy of Science, 88, 11426-11430.

[46] Ripley, B. D. (1994). Neural networks and related methods for classification. Journal of the Royal Statistical Society: Series B (Methodological), 3, 409-456.

[47] Schebesch, K. B., and Stecking, R.(2005). Support vector machines for classifying and describing credit applicants: detecting typical and critical

regions. Journal of the Operational Research Society 56, 9, 1082-1088.

[48] Shin, K. S., Lee, Y. J. (2002). Genetic algorithm application in bankruptcy prediction modeling. Expert System with Applications, 23, 321–328.

[49] Sinkey, J.F., (1975). A multivariate statistical analysis of the characteristics of problem banks. Journal of Finance 30, 21–36.

[50] Steenacker, A., and M.J. Goovaerts, (1989), A Credit scoring model for personal loans, Insurrance: Mathematics and Economics 8, 31-34.

[51] Thomas, L.C., (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting 16, 149–172.

[52] Tibshirani, R., LeBlanc, M. (1992). A strategy for binary description and classification. Jour-nal of Computational and Graphical Statistics, 1, 3-20.

[53] Tseng, F.M., Lin, L., (2005). A quadratic interval logit model for forecasting bankruptcy. Omega 33, 85–91.

[54] Ture, M., Kurt, I., Kurum, A. T., Ozdamar, K. (2005). Comparing classification techniques for predicting essential hypertension. Expert Systems With Applications, 29, 583-588.

[55] V. S. Desai,J. N. Crook, G. A. Overstreet, (1996). A comparison of neural networks and linear scoring models in the credit union environment. European Journal Operational Research 95, 24-37.

[56] V. S. Desai,J. N. Crook, G. A. Overstreet. (1997). Credit scoring models in the credit union environment using neural networks and genetic algorithms. IMA J. Math. Appl. Business and Industry.

[57] West, D. (2000). Neural Network Credit Scoring Models. Computers Operations Research 27 (11-12): 1131-1152.

[58] West, R.C., (1985). A factor analytic approach to bank condition. Journal of Banking and Finance 9, 253–266.

[59] West, P. M., Brockett, P. L., Golden, L. L. (1997). A comparative analysis of neural networks and statistical methods for predicting consumer choice. Marketing Science, 16, 370-391.

[60] Yoon, Y., Swales, G., Jr., Margavio, T. M. (1993). A comparison of discriminant analysis versus artificial neural networks. Journal of the Operational

Research Society, 44, 51-60.

[61] Yugal kumar G. Sahoo (2012) Analysis of Parametric Non Parametric Classifiers for Classification Technique using WEKA. I.J. Information Technology and Computer Science,7, 43-49.

Books

[62] Cristianini, N., Shawe-Taylor, J. (2000). An introduction to support vector machines. Cambridge: Cambridge University Press.

[63] Koza, J. R. 1992. Genetic Programming On the Programming of Computers by Means of Natural Selection. Cambridge, MA: MIT Press.

[64] Golgberg, D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning,reading. Boston, MA: Addison-Wesley.

[65] Bauer, R. J., 1994. Genetic algorithms and investment strategies. New York, NY: Wiley.

[66] Raiffa, H., Schlaifer, R. 1961. Applied Statistical Decision Theory. Boston: Harvard University Press.

[67] Altman, E.I.; Avery, R.B.; Eisenbeis, R.A.;Sinkey,Jr,J,F.; 1981, Application of Classification Techniques in Business and Finance, JAI Press Inc.

[68] Finney, P. J. 1952. Probit Analysis, Cambridge, MA: Cambridge University Press.

[69] Holland, J. H. 1975. Adaptation in natural and artificial system. Ann Arbor: University of Michigan Press.

[70] M.T. Hagan, H.B. Demuth M. Beale. 1996."Neural Network Design", Thomson Learning USA.

[71] D. Michie, D.J. Spiegelhalter, C.C. Taylor (1994) Machine Learning, Neural and Statistical Classification.

A

Appendix 1

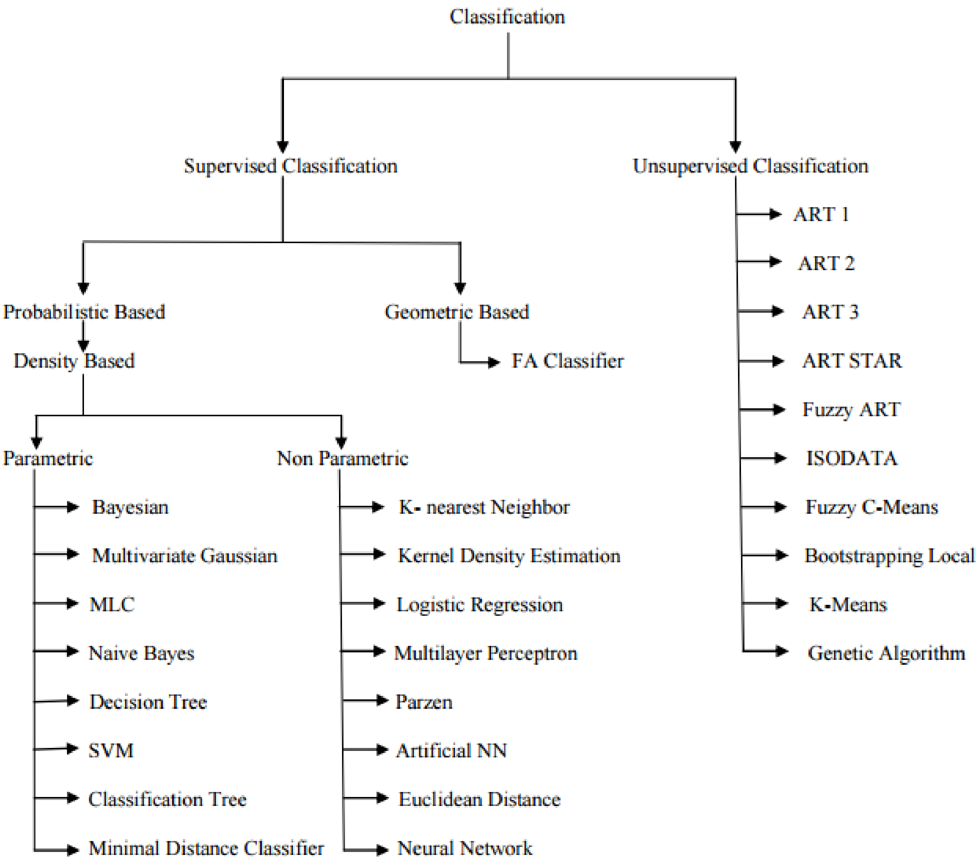
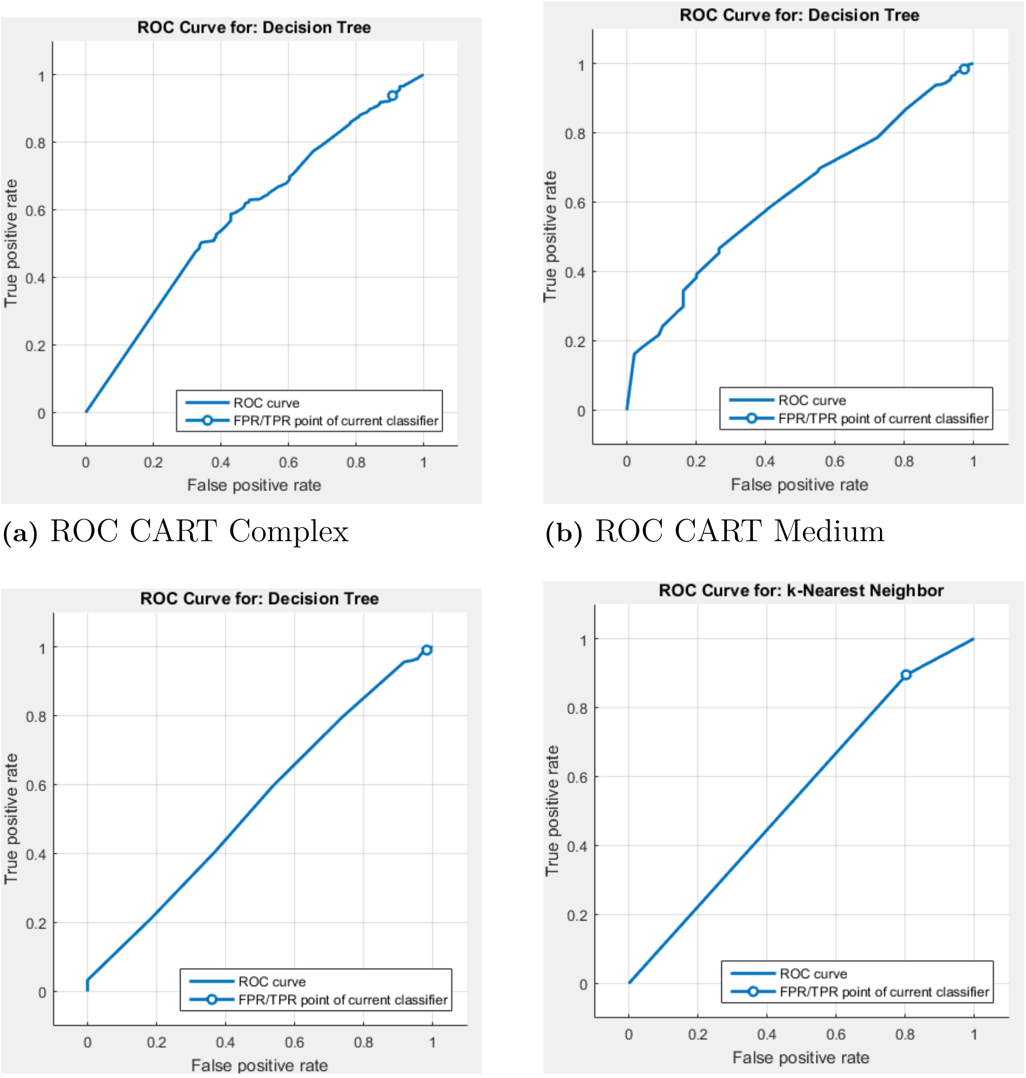


Figure A.1: Overview of machine learning techniques

B

Appendix 2 - ROC Germany



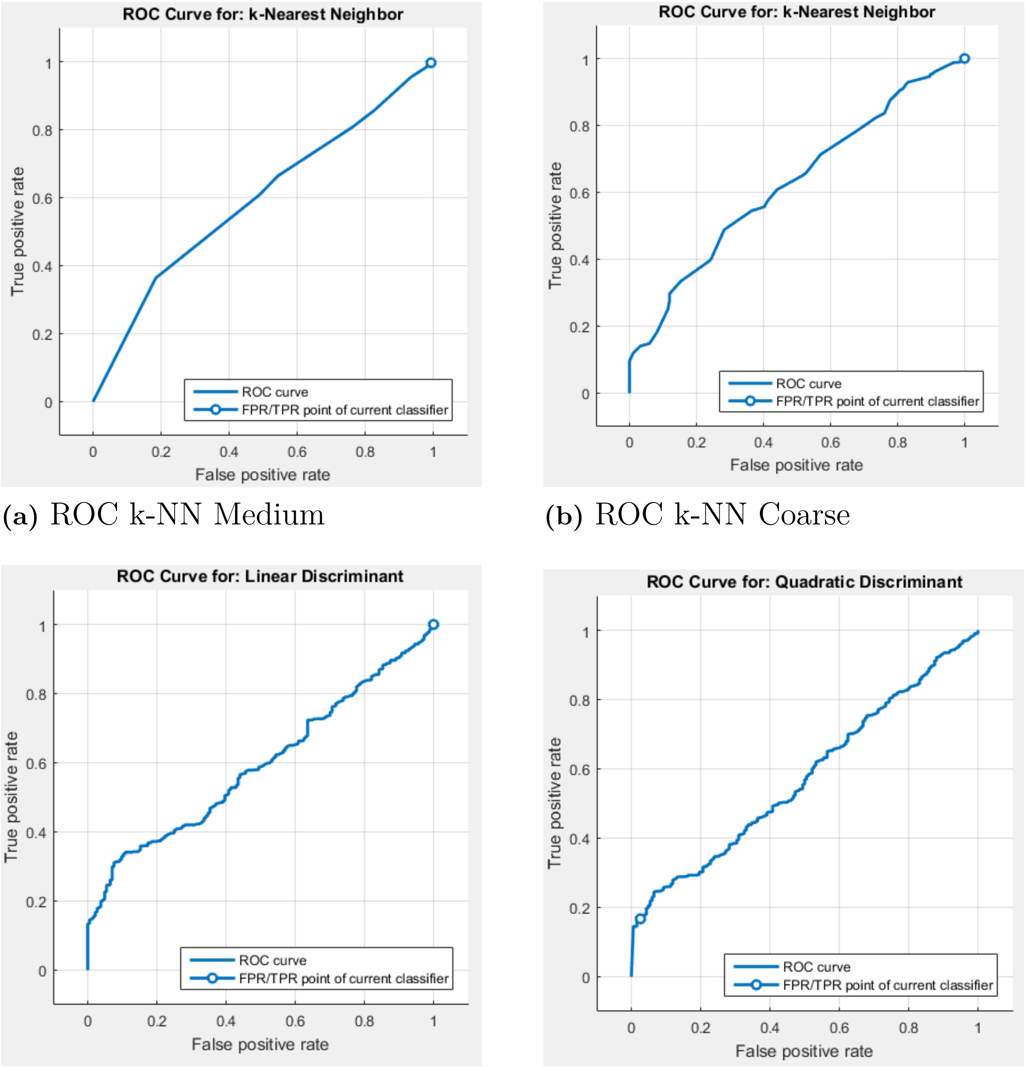
(a) ROC CART Complex

(b) ROC CART Medium

(c) ROC CART Simple

(d) ROC k-NN Fine

Figure B.1: TROC curves for the models.



(a) ROC k-NN Medium

(b) ROC k-NN Coarse

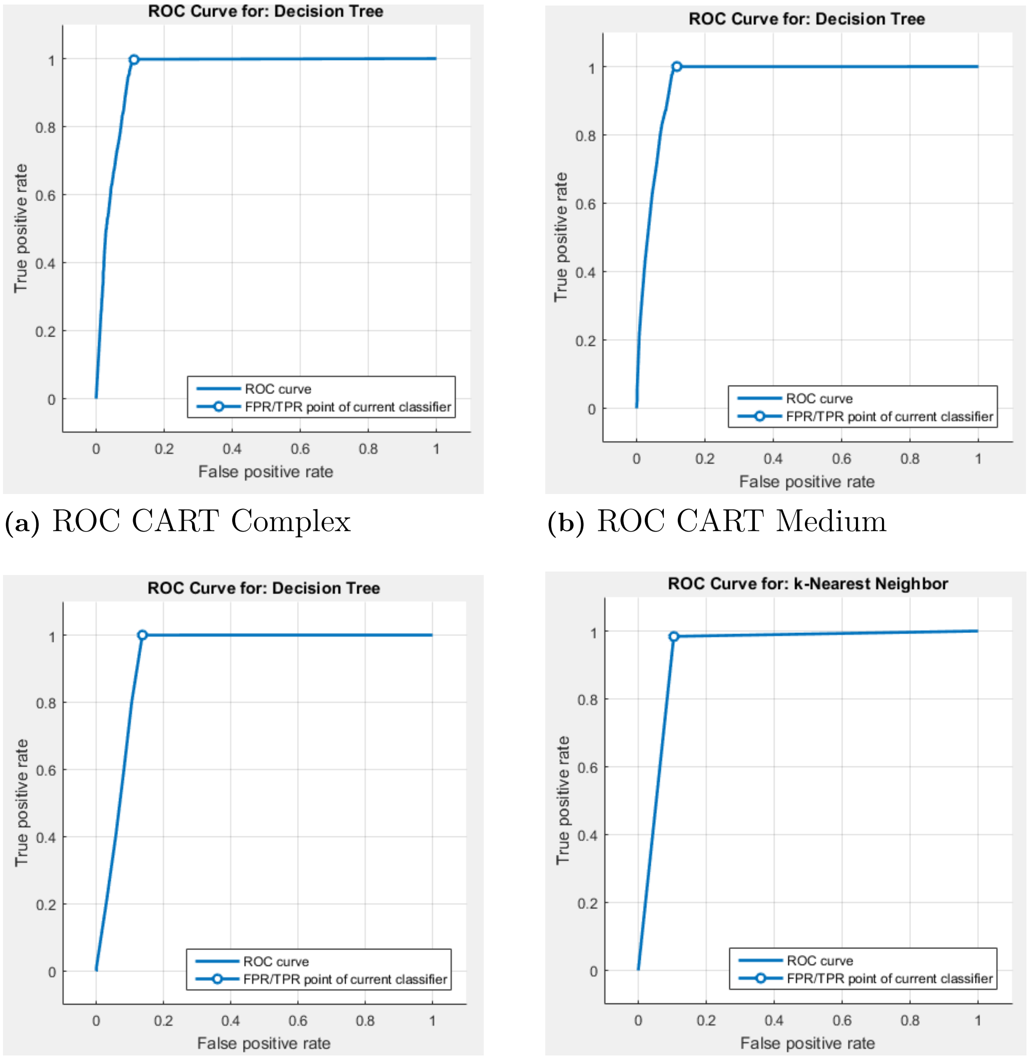
(c) ROC LDA

(d) ROC QDA

Figure B.2: ROC curves for the models.

C

Appendix 3-ROC France



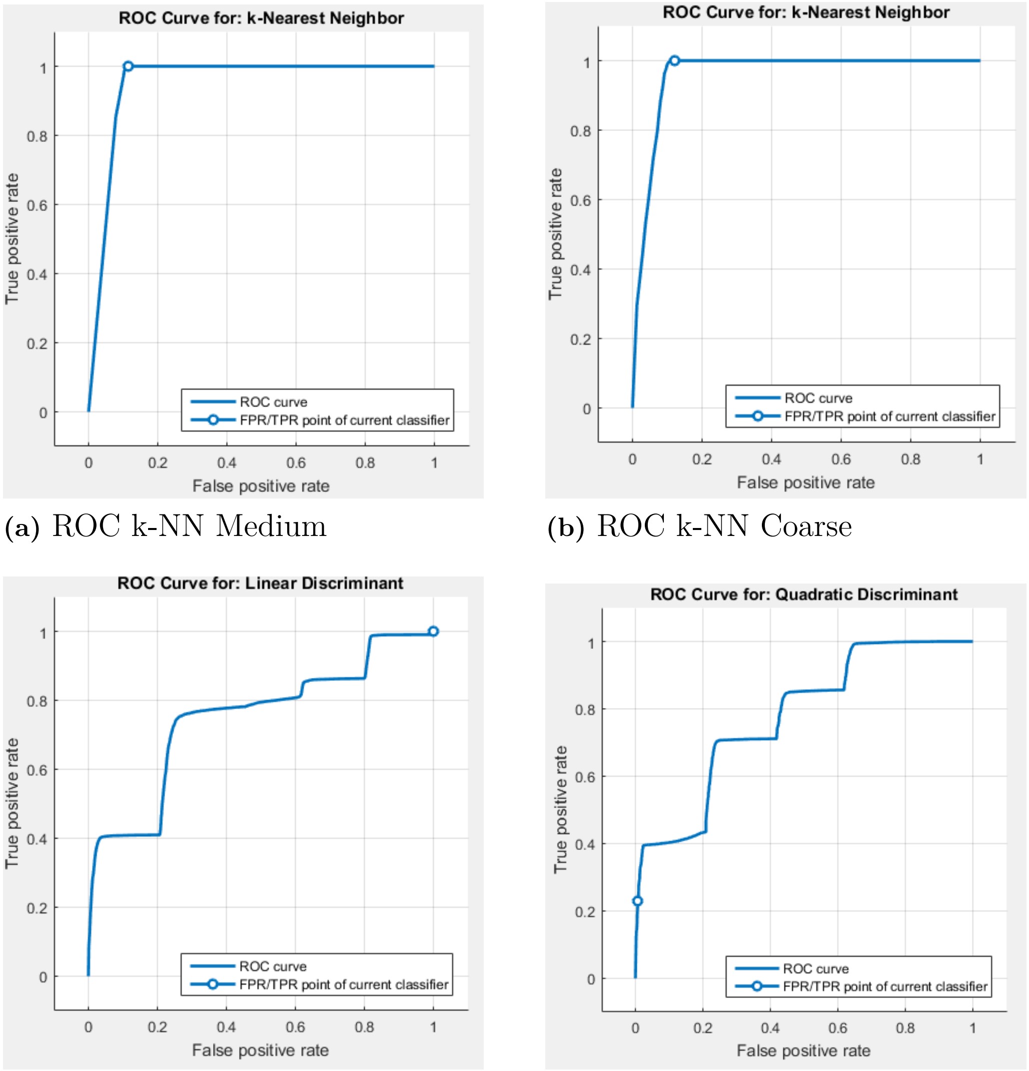
(a) ROC CART Complex

(b) ROC CART Medium

(c) ROC CART Simple

(d) ROC k-NN Fine

Figure C.1: TROC curves for the models.



(a) ROC k-NN Medium

(b) ROC k-NN Coarse

(c) ROC LDA

(d) ROC QDA

Figure C.2: ROC curves for the models.