Mikko Laulajainen

Case Study on Utilizing Machine Learning in Corporate Default Risk Prediction

A practical Implementation to Credit Risk Management Process

School of Technology and Innovations
Master of Science in Economics and Business Administration
Master's Programme in Industrial Management

Vaasa 2022

UNIVERSITY OF VAASA
Faculty of technology
Author: Mikko Laulajainen
Title of the Thesis: Case Study on Utilizing Machine Learning in Corporate Default Risk Prediction
Degree: Master's Degree
Programme: Industrial Management
Supervisor: Petri Helo
Year: 2022 **Pages:** 127

ABSTRACT:

The purpose of the case study is to create an in-house corporate default risk prediction model that outperforms the external corporate credit rating which the case company is currently using for this purpose. In addition, the study sets the framework for implementing the model into current system architecture and credit risk management process. The study consists of literature review and empirical analysis where the default prediction models are built and tested and the proposal for implementing the model into case company's system architecture and processes is given.

The data used in this study consists of historical financial figures & ratios, payment behaviour information and other background information of 2471 Finnish companies from period 2009-2017 of which 22,6% defaulted during this period. MissForest method was used in imputation of the missing values. The models used in this study are Multivariate Discriminant Analysis, Logistics Regression, Random Forest, CART, AdaBoost, Support Vector Machine and Neural Network. The dataset was split with 70/30 ratio to training and test set and 10-fold cross validation was used in training, feature selection and hyperparameter optimization for each model. Model performance was also tested over a two-year time horizon.

The models' performance was measured with ROC AUC & PR AUC and Brier Score. All the models overperformed the external credit rating with the selected metrics. The best performing model was the black box model Adaboost and the best performing white box model was the logistic regression with LASSO method used for the predictor variable selection.

KEYWORDS: Credit Risk Management, Default Risk, Machine Learning, Default Risk Prediction

VAASAN YLIOPISTO
Tekninen Tiedekunta
Tekijä: Mikko Laulajainen
Tutkielman Nimi: Case Study on Utilizing Machine Learning in Corporate Default Risk Prediction
Tutkinto: Maisteriohjelma
Oppiaine: Tuotantotalous
Työn Ohjaaja: Petri Helo
Vuosi: 2022 **Sivumäärä:** 127

TIIVISTELMÄ:

Tutkielman tarkoituksena on luoda kohdeyritykselle sisäinen luottoriskin ennustemalli, joka ennustaa asiakasyirtysten luottoriskiä tarkemmin kuin yrityksen käyttämä ulkoisen luottoriski-luokittajan luokitus. Lisäksi työsä kuvataan miten malli voidaan implementoida kohdeyrityksen järjestelmä-arkkitehtuuriin ja riskienhallintaprosessiin. Tutkielma koostuu kirjallisuuskatsauksesta, jossa esitetään luottoriskmallinuksen kehitys ja yleisesti käytetyt mallit, empiirisestä analyysistä, jossa mallit rakennetaan sekä testataan. Lopuksi asetetaan viitekehys ja annetaan suositukset mallin implementoinnille kohdeyrityksen järjestelmäarkiterhuuriin ja prosesseihin.

Aineisto koosutuu 2471:den suomalaisen yhtiön taustatiedoista, historiallisista taloudellisista tunnusluvuista, sekä maksukäyttäytymistiedoista vuosilta 2009-2017. Yrityksistä 22,6% aiheutti asiakasyritykselle luottotappiota kyseisellä aikajaksolla. MissForest imputaatiota käytettiin puuttuvien arvojen täydentämisessä. Kirjallisuuskatsuaksen pohjalta valitut mallitut tutkielmassa ovat monimuutuja erotteluanalyysi (eng. Multivariate Discriminant Analysis), logistinen regressioanalyysi, satuinnaismetsämalli, CART-päätöspuumalli, Adaboost-päätöspuumalli, tukivektorikone sekä neuroverkko. Aineisto jaettiin opetus ja testiosioihin suhteessa 70/30 ja 10-fold ristiinvalidointia käytettiin mallien luontiin, muutujen valintaan sekä hyperparametrien optimointiin. Malliien toimivuus testattiin myös kahden vuoden ajanjaksolla.

Mallien toimivuutta mitattiin ROC- ja PR-käyrien alle jäävän pinta-alalla, sekä Brier score-pisteytyksellä. Kaikki mallit erottelivat luottoriskin todenäköisyyden tarkemmin, kuin verrokiksi asetettu ulkoinen luottoriskiluokitus. Paras malli edellä mainituilla mittareilla mitattuna oli mustaksi laatikoksi kuvailtava päätöspuumalli Adaboost. Paras helpommin ymmärrettävä malli oli logistinen regressio, jossa ennustemuutujien valinta on tehty LASSO-menetelmällä.

AVAINSANAT: Luottoriskin hallinta, luottoriski, koneoppiminen, luottoriskin ennustaminen

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ACRONYMS

AUC: The Area Under the Curve
BBR: Bureau of Business Research
BC: Brier Score
BCBS: Basel Committee on Banking Supervision
CART: Recursive Partitioning Algorithm
EAD: Exposure at Default
EL: Expected loss
LogR: Logistic Regression
LGD: Loss given default
MDA: Multivariate Discriminate Analysis
ML: Machine Learning
MSE: Mean Square Error
PR: Precision Recall
PD: The Probability of Default
ROC: The Receiver Operating Characteristic curve
SVM: Support Vector Machine
UL: Unexpected loss

1 Introduction

Risk management became an important concern for organizations after the Great Depression in the 1930s and has since become one of the main areas of interest in the field of financial research. The financial crisis of 2008 followed by introduction of Basel III regulation to promote stability in the international financing system only increased the need for credit risk modelling in the financial institutions. The need for credit risk modelling does not only derive from regulation but is an area of interest for all organizations that sell products or services with credit. The recent COVID-19 pandemic, which impacted all companies globally, affected SME businesses to even larger extent and has increased the importance for functioning default risk prediction model for SME businesses (Adian et al. 2020, Ciampi et al. 2021). The most recent shock, Russia's invasion of Ukraine, and significant increase in inflation driven by the increase in commodity prices, has further increased the need for default prediction models that take also macro-economic factors into account.

This thesis is a case study which aims to improve the credit risk management process of a Finnish retail company and to create a superior default prediction model to currently used externally sourced credit rating. A quantitative analysis of the probability of default of Finnish companies from various industries and sizes is conducted by utilizing commonly used statistical and machine learning methods in default prediction. Selection of the models is done based on previous academic research. The performance of the selected models is evaluated against the external credit risk rating. The data used in this thesis consists of company characteristics, historical financial statements, internal and external payment behaviour and historical losses. In addition, the thesis sets the framework for implementing the model into case company's system architecture and credit risk management process.

1.1 Targets of the study

The purpose of the study is to create an in-house default prediction model for the case company, and to evaluate whether introduction of internal variables can help the model to outperform currently utilized external credit rating. The current rating-based risk assessment and risk monitoring requires a lot of manual work to assess whether the lowest ratings, with highest probability of default, require immediate actions. Therefore, this study aims to find a superior model with better classification performance, to decrease case company's credit losses.

The requirements for set for the model are *transparency*, *accuracy,* and *cost effectiveness*. To assess the correctness of the model and to explain the reasons behind credit decisions for customers, the data involved must be *transparent* to the end users. *Accuracy* is required, to ensure that the model adds value to new processes and gives minimal amount of false alarms. Cost always plays a part, when assessing the business case of a project and therefore, the model must be *cost effective.* Acquiring new data from external sources can be expensive and therefore the model should utilize data which is already available to the case company. In addition, the cost of implementing the model into current system architecture and credit risk management process should be kept low.

1.2 Definition of the research problem

The main research questions are:

Can an in-house machine learning default risk prediction model, which utilizes both, internal and external predictor variables outperform the currently utilized external credit risk rating?

Which model performs the best?

The following sub-question is also addressed:

How to implement the model effectively to the credit risk management process?

1.3 Delimitations

The purpose of the thesis is to create a working and understandable model by utilizing best practices in credit risk modeling and sets the framework for the model in case company's credit risk management process. The target of the literature review is to recognize models that have been successfully utilized in default prediction in the previous research and the scope of this study does not give a full and extensive review of all models introduced in academic research. The predictor variables used in this study are limited as only data that is currently available to the company is utilized. For the same reason, the study uses a data sample from years 2009-2017 for training and testing. It is recommended to validate the results of this study with a larger and newer dataset before deploying the model into use.

1.4 Structure of the study

Chapter 1: Introduction

In the first chapter, the background and targets for the study are explained. The research questions, delimitations are set, and the structure of the study is presented.

Chapter 2: Theoretical framework and previous research

The second chapter describes the theoretical framework and the main concepts, namely definition of the defaulting and credit risk components and gives introduction to the evolution of default prediction modelling and previous research.

Chapter 3: Research Methods

In this section, the research method, selected models and evaluation methods for the performance of the models are introduced.

Chapter 4: Empirical Analysis

In this chapter, the case company, its credit risk management process, and the current challenges in the process are introduced. The baseline metrics of the external credit rating are set, and the dataset and financial variables are presented along with quantitative analysis. Finally, the problem of missing data is solved with selected imputation method.

Chapter 5: Model Development and Performance

In this chapter the models are created and optimized and the performance metrics for each model are presented and discussed.

Chapter 6: Model selection

In this chapter, the results presented in previous chapter are discussed further and models are compared against each other and the set baseline. Finally, the recommendation for model selection is given.

Chapter 7: Fitting the model into ERP Architecture and Credit Risk Management Process

Recommendation of how to implement the model into current ERP architecture and Credit Risk Management process is given in this chapter.

Chapter 8: Conclusions and Summary

In the last chapter, a summary of the results and the limitations of the study are presented, and all the research questions are answered. Finally, recommendation for future research areas is given.

2 Theoretical Framework and Precious Research

To discuss supply credit risk modeling, the basic concepts of defaulting and credit risk modeling must be defined. This section provides general definitions for the key concepts, evolution of default prediction modelling and presents commonly used statistical methods. Based on this review, the models used in the study are selected.

2.1 Definition of Defaulting and Credit Risk Components

Although, default is universally acknowledged term and commonly used in academic research of credit risk modeling, there is no standard definition set. In the early studies, the terms “failing firms” or “business failure” were used. In many studies, failure was defined as an actual bankruptcy or liquidation; while in others failure is defined as suffering of financial stress or inability to fulfill the financial obligations (Karels and Prakash, 1987) and in some research failure is not clearly defined. (Lim, 2012)

Since the late 90s, the majority of the research is focused on estimating the probability of default (PD) and the loss given default (LGD), both of which are in fact the two key risk parameters in the internal rating based (IRB) approach defined by the Basel II accord. (Crouhy et al. 2000). The Basel II accord, and later extended and partially superseded by Basel III, is an international banking standard that controls how much capital the banks are required to hold as a guard against financial and operational risks. (The Federal Reserve Board, 2006; Bank for International Settlements, 2009). The accord is set by worldwide standard-setter, Basel Committee on Banking Supervision (BCBS). BCBS’s purpose is to strengthen the regulation, supervision and practices of financial institutions (BCBS 2006). Although, the case company does not operate within the scope of Basel II/III accord, the widely acknowledged credit risk components, can be derived from the accord.

According to Basel II/III legislation (BCBS, 2006), default occurs if both or either of the following scenarios take place.

1. “The credit institution considers that the obligor is unlikely to pay its credit obligations to the institution in full, without recourse by the credit institution with actions such as realizing security (if held). “ (BCBS, 2006),

2. “The obligor is past due more than 90 days on any material credit obligation to the banking group. Overdrafts will be considered as being past due once the customer has breached an advised limit or been advised of a limit smaller than current outstandings.“ (BCBS, 2006),

Many articles, such as the studies of Chalupka and Kopecsni (2008), Bonfim (2009), and Schmit (2004) follow the second part of the Basel II/III definition, but some other research, such as the articles from Agarwal and Taffler (2008) and Grunert and Weber (2009) consider default to occur only in case of bankruptcy. It should be taken into account, that some obligors may pay their debt even after 90 days. In some cases, the overdue debt may be a result of bad payment discipline, rather than true insolvency. In Chapter 4.4 we will discuss what approach this thesis takes on the definition of defaulting.

Expected loss can be divided into three components:

Probability of default (PD) is the probability of default of a counterparty. PD is generally estimated by reviewing the historical defaults of other counterparties with similar characteristics. In the Basel II/III accord (BCBS, 2006, 2017) it is defined as the probability of default over a one-year period, but other estimations can also be done over shorter or longer time periods. In terms of retail business, the maturity of the loan is essentially the payment term of the purchase on credit. This needs to be considered retailer usually has many possibilities to react proactively to the evident defaulting event by e.g. decreasing the line of credit, asking for collaterals or by requesting a prepayment for new orders.

Loss given default (LGD) is them monetary loss that occurs due to default of a counterparty. (BCBS, 2006, 2017) Lenders can protect themselves by requesting a collateral, by holding credit derivatives as a security. (Ong 2007). Another commonly used tool in retail business to minimize LGD is using credit insurances. According to Berne Union (2021), a non-profit trade union of the global export credit and investment insurance

industry, the worldwide insured exposure was approximately 2.8 trillion US dollars in 2021. In case of a protective measures taken by the lender, the LGD is the value which is not covered by the collateral, guarantor, or credit insurance (Ong, 2007).

Exposure at default (EAD) is the amount outstanding at the time of default when the value of collateral is deducted from the outstanding amount. It is also called as the current exposure (BCBS, 2006, 2017)

These three components then form the *expected losses (EL)*. EL can be estimated with the aforementioned components: (The Federal Reserve Board, 2006)

$$Expected\ loss\ (EL) = PD \times LGD \times EAD \quad (1)$$

In addition, unexpected losses may occur, in addition to the expected losses. Unexpected losses represent potential losses over and above the expected loss amount. Unexpected loss can be defined by using a specific percentile threshold of the probability distribution. Example of unexpected losses is presented in Figure 1

Figure 2. Linear SVM (Joshi 2020). at loss distribution threshold of 99.9th percentile. (The Federal Reserve Board, 2006)

$$Unexpected\ loss\ (UL) = f(PD, LGD, EAD) \quad (2)$$

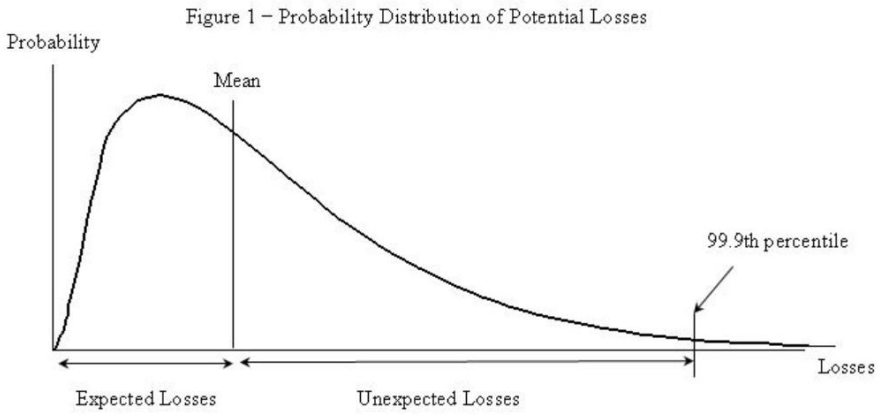

Figure 1. Probability Distribution of Potential losses (Basel Committee on Banking Supervision 2006).

This study focuses on predicting the PD, as it can be considered as the most important factor in preventing credit losses.

2.2 Early studies of corporate failures

From 1930's to mid-60's, most of the academic research studied the relationship and differences in financial figures and ratios of successful and failing firms. One of the earliest analysis of financial ratios of was published in 1930 by the Bureau of Business Research (BBR), a year after the beginning of the Great Depression. This paper studied 24 ratios of 20 failed companies. Results suggested that eight out of the 24 ratios have similarities between the companies that failed and could therefore indicate a higher default risk. The most valuable indicator highlighted by the study was the working capital to total assets.

Similarly, FitzPatrick compared 13 ratios of 19 failed and 19 successful in year 1932. The article suggested net worth to debt and net profit to net worth as the most important ratios for failure prediction, while the current and quick ratio did not predict the failure of a company as well.

In follow-up study to the BBR's publication, Smith and Winakor (1935) analysed ratios of 193 bankrupt firms from various industries and suggested that that Working Capital to Total Assets is a better predictor of financial problems than Cash to Total Assets and the Current Ratio.

Merwin's article published in 1942 focused on comparing successful and failing small manufactures. The key finding of the study was, that the failing firms start to display signs of weakness already four or five years before bankruptcy. The paper found that weak Net Working Capital to Total Assets, Current Ratio and Net Worth to Total debt may indicate business failure.

Chudson published an article in 1945 that focused on finding patterns of financial structure to determine if there is a "nominal" pattern for profitable firms. The concluded that within industry, size and profitability groups, a clustering of ratios can be found, but there is not a one decisive structure for profitable firms. These results are significant for

credit risk modeling, as the results suggest that as industry-specific models may perform better than general models.

2.3 Evolution of default prediction models

After the early studies that suggested that certain financial figures and ratios and changes can indicate a potential failure, there was an interest to create models that assess the default risk theoretically and practically instead just assessing the risk by relying on subjective analysis of credit experts. (Beaver, 1966; Altman 1968). The first credit scoring models were proposed by Beaver (1966) and Altman (1968). These studies named firm-specific financial ratios to predict the probability of bankruptcy.

2.3.1 Univariate Discriminate Analysis (UDA) model

An important milestone the academic research of default prediction was achieved in 1966, when Beaver published article called “Financial Ratios as Predictors of failure”. In this study, Beaver analysed 158 industrial companies of which 79 failed by using Univariate Discriminant analysis. Data was collected from period 1954 to 1964 with one to five financial statements from each company before the year of bankruptcy. The study used a very simplistic univariate approach and compared the mean values of each financial ratio to the ratios of individual companies. According to the study, Net Income to Total Debt ratio had the highest accuracy of 92% in predicting the of the company one year before the failure. The second-best ratio was the Net Income to Sales with accuracy of 91%. Cash Flow to Total Debt, Net Income to Net Worth and Cash Flow to Total Assets were also good indicators with accuracy of 90%.

There were limitations to this study as Beaver did not take into account the distribution of different variables and the failing and non-failing companies may have represented different populations, but more importantly, Beaver suggested that future research should consider analysing multiple ratios simultaneously as multivariate models may have higher predict default better than analysing only single ratios. This suggestion started the evolution of bankruptcy prediction models. (Gissel; Bellovary et al. 2007)

2.3.2 Multivariate Discriminate Analysis (MDA) Model

Altman published the first linear multivariate discriminate analysis for default prediction model in 1968. By using multivariate discriminant analysis, Altman developed a model called the Z-score to predict the bankruptcy of manufacturing firms. The method was chosen due to its previous success in consumer credit evaluations and investment classifications. Chosen sample consists of 66 SMEs with assets ranging from one to 25 million dollars and half of which had bankrupted within the period of 1946-1965 and the other half was still functional in year 1966. (Altman 1968)

Altman chose 22 variables based on their popularity and potential predicting power for bankruptcy modeling in previous research. Of these 22 variables, a combination of 5 variables was chosen for the model, not only based on the significance of the individual variable, but on the basis how well these variables predicted bankruptcy as a combination. (Altman 1968)

The following Z-score formula was proposed by Altman (1968):

$$Z = 0.12X1 + 0.14X2 + 0.033X3 + 0.006X4 + 0.999X5, \quad (3)$$

where,

X1 = Working Capital to Total Assets.
X2 = Retained Earnings to Total Assets.
X3 = Earnings Before Interest and Taxes to Total Assets.
X4 = Market Value Equity to Book Value of Total Debt.
X5 = Sales to Total Assets.

For the sample used in the study, the model was able to predict bankruptcy one year before failure with an accuracy of 95%. However, the accuracy dropped substantially for longer prediction periods. The accuracy of two years before the failure for the sample is 72%, and only an accuracy of 48%, 29% and 36% for three, four and five years before the failure, the hold-out sample accuracy for this model however was only 79%. However, the results show that the created Z-score model was capable to differentiate healthy companies from failing companies, even in cases where individual variables showed poor performance over the company. Altman (1968)

The model has mainly been criticised due to the assumption, that the individual variables are normally distributed. (Eisenbeis, 1977; Rosenberg & Gleit 1994) However, empirical study of Reichert et al. (1983) showed, that normal distribution is not a critical limitation in practise. The main advantage of multivariate discriminant analysis is that the individual weights show the contribution of each predictor variable making the model easy to interpret.

In 2013, Altman revisited the Z-score model and introduced a new bankruptcy prediction model called ZETA-model. The model included additional market variables to the Z-score. Based on the accuracy testing, the ZETA-model outperformed the original Z-score, and the hold-out sample accuracy was found to be 93%. (Callaghan et al., 2015, 24-30)

2.3.3 Logistic Regression

Use of linear regression in credit risk prediction started in the late 1960s. One of the first study was conducted by of Ewert (1969) with classification accuracy of 82%. The use of logistic regression was replaced in early 1980s by logistic regression in the studies of Ohlson (1980) and Wiginton (1980), who concluded that it performs better than discriminant analysis. Similar findings have been also confirmed in the later studies such as in the study of Altman and Sabato in 2013 in their study of default prediction for SME companies, Cultrera's and Brédart's (2016) study of SMEs in Belgium, to name a few.

Logistic regression (LogR) is considered as a superior model in statistical analysis, as many of the conceptual and computational challenges of linear regression, such as possibility of negative possibility and possibility with larger than one, can be taken into account in the model. It also has several advantages over discriminant analysis, normal distribution of the input variables is not required and therefore also qualitative variables can be included in the model. (Henley & Hand, 1997)

2.4 Machine learning methods in default prediction

Thenumber and complexity of models have increased along with increased computational power since the models published by Beaver, Altman and Ohlson. In the early 21st

century, the most commonly research methods in academic papers were multivariate discriminant analysis (MDA), logistical regression (logit and probit analysis), and neural networks. (Gissel et. al. 2007). One of the main advantages of machine learning (ML) models the ability to find patterns in the data more efficiently than statistical models due to ability to handle non-linear relationships in the data. In practise, these models can however be difficult to be interpreted by humans. (Van Liebergen, 2017)

Literature review of Leo et al. (2019) concluded, that ensembled decision trees, Support Vector Machines and Neural Networks are the most common ML -methods for corporate default risk predictions in the financial sector and generally outperform other default prediction models.

In this chapter the characteristics of commonly used ML models are discussed in further detail and pros and cons for each model is highlighted based on previous research.

2.4.1 Neural Networks

Neural networks became popular in the academic research in 1990's. Neural networks are designed to emulate the process of the human brain. Neural networks can be described as multi-stage information processing. where at each stage the hidden correlations of the predictor variables are identified. The complexity of the model makes it extremely difficult to interpret and the model can therefore be considered as a black box model. (Anandarajan et al., 2004).

Although this more complex model has been utilized to improve the prediction accuracy of credit risk modelling, a comparison of the prediction accuracy in the academic research reveal, that the average overall classification performance of neural network models is similar to logistic models (Aziz & Dar, 2006).

2.4.2 Support Vector Machines

Support Vector machine was first introduced by Vapnik in 1999 and is based on the structural risk minimization principle from computational learning theory. The SVM aims to categorize datapoints in a high-dimensional space. After finding a separator between the categories, then the data is transformed so that the separator with minimal number of used datapoints can be drawn as a hyperplane. The observations in the dataset are

classified based on which side of the hyperplane they are. The model aims to maximize the class boundaries' distance with support vectors. Linear and non-linear data can be handled by SVM with different kernel functions. (Hsu et al., 2004)

Figure **2** represents a linear SVM. The dots represent the observations and class in the dataset, the solid line is the hyperplane and the dotted lines represent the boundaries between the classes. (Joshi, 2020)

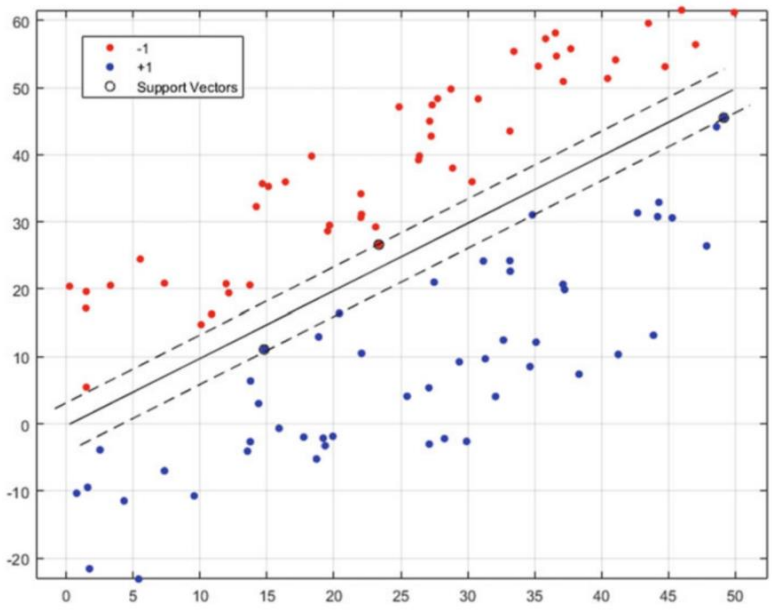

Figure 2. Linear SVM (Joshi 2020).

Many studies have confirmed SVM as a successful method for default prediction. Study of Fan & Palamiswami (2000) concluded that SVM outperformed neural networks and linear discriminant classifiers, Moula et. al. (2017) showed that the SVM model is marginally superior to CART, Yu et al. 2008 reported that SVM performed slightly better than logistic regression, Shin et al. (2005) concluded that SVM performed better a back-propagation neural network and Chen (2011) found that in comparison to other prediction models, the SVM had high accuracy and performed well for both, short and long-term predictions.

2.4.3 Decision trees

Decision trees utilize hierarchical decision-making process, which is similar to human behavior in real-life decision-making process and can be used for both, classification and regression. One key advantage of decision tree is that trees can also interpret categorical data. (Joshi 2020, 53-63)

Classification & Recession tree (CART), also known as Recursive Partitioning Algorithm, is a data mining method, that utilizes decision trees in classification. Study by Frydman et al. (1985) concluded, that CART outperformed MDA in most sample and holdout comparisons and is good at giving additional, easily interpreted information of the relationship of the predictor variables.

Ensembled decision trees such as Random Forests and AdaBoost are more complex models, that utilize either boosting or bagging techniques. These models function by building multiple trees and then adapting the predicting power of these trees into one model to improve model performance. Boosted decision trees are trained in sequence, where the models learn from mistakes of the previous models. Bagged trees are conducted by first training individual models with a random sample subset before aggregating the results into one model.

Random forest was first introduced by Breiman in 2001 and it utilizes a bagging algorithm. One key advantage of bagged trees is that they perform well with outliers, as single tree does not affect the whole model significantly. Adaboost is an ensembled decision tree, which utilizes adaptive boosting algorithm, and was first introduced by Freund and Schapire in 1996. Adaptive boosting works by combining weak classifiers to create a single strong classifier. The challenge with the boosted trees is, that due to sequential process, the trees cannot be built parallelly and are therefore slower to run in comparison to bagged trees (Joshi, 2020, 53-63).

2.4.4 Other models

The field of default prediction modelling is constantly developing and beside the aforementioned models, there are multiple other, and more complex models that have been successfully used in default prediction. For example, Shetty and Vincent (2021) used

graph theory in corporate default prediction in Indian market, Lee et al. (2021) used Graph convolutional network for predicting defaults for private borrowers, Li et al. (2022) used blending method to fuse Random Forest, Logistic Regression and CatBoost into one model. The connective factor between these models seem to be, that the achieved prediction performance improvements are marginal in comparison to the added complexity in interpreting the models. The ease of interpretation might explain why MDA and logistic regression are still some of the most used models in the field of default prediction.

3 Research Methods

3.1 Case Study

The purpose of the case study is to create an automated credit prediction model to forecast the probability of default of the case company's corporate customers and give proposal how to implement the model into the credit risk management process and system architecture. The model is created by using R and its prediction power is compared to the currently used credit rating. The R code is designed so, that it can be implemented into the current system architecture with minor changes. This means that the model needs to be able to be scheduled to run independently and handle imperfections in the process e.g. missingness of the data.

Inputs for the model are case company's industry, financial key figures and ratios from financial statement, internal and external payment behavior metrics. The output of the model is the predicted probability of default. In the case study, the dataset is introduced, imputation method for the missing data is selected, the models are trained and tested against the baseline set by the external credit rating and finally, a proposal of how to implement the model into current system architecture and credit risk management process is given.

3.2 Used Algorithms

The selected models based on the literate review are Multivariate Discriminant Analysis (MDA), Logistics Regression, Random Forest, AdaBoost, Support Vector Machine (SVM) and Neural Network. These have all been successfully used in the field of default prediction. The selected models and the performance in the previous studies were presented in chapters 2.2 and 2.3.

3.3 Evaluation and validation of the models

Validation of the model performance is essential for development process of a default risk prediction model. Validation is used to compare the performance of the developed models. There are two aspects to the performance of the model: discriminatory power and speed. The discriminatory power of the default prediction model refers to model's fundamental ability to differentiate default and non-default companies. Depending on the use case and available computational power, also the speed of the model can be as important, however usually the evaluation is done by evaluating the discriminatory power, as models are rarely so complex that the amount of computational power would become an issue. (Kubat, 2017, 211-229). In the scope of this thesis, the focus is on the discriminatory power, as there is sufficient computational power available in the proposed cloud-based system architecture.

In this study, we confirm in chapter 4.3, that the dataset is not balanced as there is more non-default companies than defaulted companies in the dataset. Therefore, the selected metrics need to consider the imbalance of the dataset. Some traditional metrics such as threshold metrics accuracy and error do not work with imbalanced dataset, as high accuracy is achievable by a model that classifies all cases to the majority class.

The machine learning model evaluation metrics for classification models can be divided to three families: *Threshold metrics*, *the ranking metrics*, and *the probabilistic metrics*. (Ferri et. al., 2009)

The threshold metrics such as accuracy, macro-averaged accuracy, mean F-measure and Kappa statistic, can be used when minimizing the number of prediction errors is critical. (Ferri et. al., 2009). These metrics are used to evaluate the model's classification performance and require a use a specific threshold in making the classification. E.g. a model could be set to predict, that all companies with probability of default of more than 50% are predicted to default in the future, if the threshold level is set to 0,5. Therefore the result of these metrics change depending on the set threshold of the model and are not therefore suitable for models that predict probabilities. Due to this limitation, these metrics are not used in this study.

The ranking metrics are used to evaluate model's ability to separate classes. Some commonly used metrics are receiver operating curve (ROC) and the ROC area under curve

(ROC AUC). Precision – Recall curve (PR) and the corresponding area below the curve (PR AUC), focuses on the performance classifying the minority class. (Ferri et. al., 2009) These metrics are classification-threshold-invariant and measures the quality of the model's predictions, do not require a set threshold and can therefore be used when evaluating a model with probability output. Both, ROC-AUC and PR-AUC are selected to be used in this study in comparing the different models.

The probabilistic metrics are used to measure the deviation of the prediction from the true probability. Some commonly used metrics are mean squared error (Brier score), LogLoss), the probability rate. These measures assess the reliability of the classifiers. These metrics measure the confidence of the model in making the predictions. (Ferri et. al., 2009) In this study, we select Brier score as one evaluation metric of the model, which can be used with unbalanced data.

In the following sections an introduction and discussion on the selected methods and metrics used to quantify the performance of the models.

3.3.1 Receiver Operating Characteristic curve

The Receiver Operating Characteristic curve (ROC) is the most widely used metric to depict the discriminatory power of the classification model. This metric was selected as performance metric in this study, as it does not rely on threshold settings, can be used to evaluate a model with probability output and works with imbalanced classifications. (Gong, 2021)

ROC represents the relation between true positive and false positive predictions of the models at different thresholds. The ROC curve is constructed by plotting the fraction of the false positive predictions on the x-axis against the fraction of the true positive predictions on the y-axis. (Kotu & Deshpande, 2014)) Figure 3 illustrates various ROC curves. The point in top left corner represents perfect model, where all cases all cases are classified perfectly, and dotted line represents a model with no prediction power.

For interpretation purposes the area under the ROC curve can be calculated (ROC AUC). A model with perfect classification has the ROC AUC of 1 and model with no discriminatory power would have an ROC AUC of 0,5. (Kotu & Deshpande, 2014) It should be noted that ROC AUC does not take into account the shape of the ROC curve. In Figure 3, the

curves A and B represent ROC curves of two different models with the same AUC measure. The steeper curve A, can be considered better in default prediction, as it allows using higher cut off to avoid false positives (predicted to default, but did not) with smaller impact of misclassifying true positives (predicted to not default, but defaulted). This is important, as misclassifying non-defaulters as defaulters, can be costly as it results in a loss of sales.

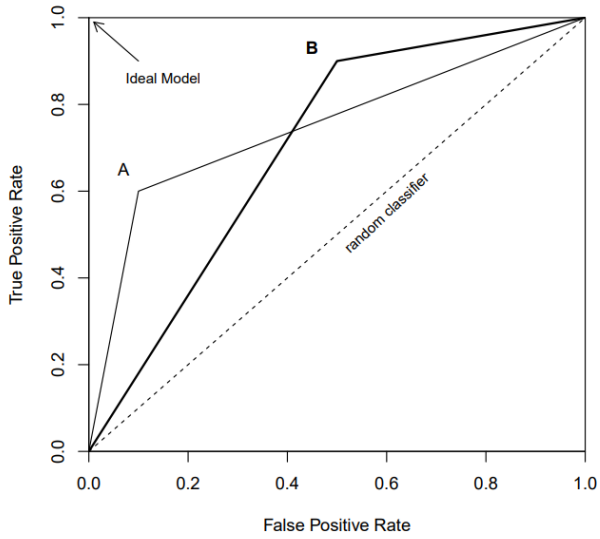

Figure 3. Example ROC curves (Branco et. al. 2016).

3.3.2 Precision-Recall cure

Precision-recall curves (PR curves) represents the relation between mode's precision and recall at different classification thresholds. Precision-recall curves are recommended for highly imbalanced datasets and can give more informative picture of models' performance than ROC curves. *Precision* is the ratio of correct positive predictions to the total positive predictions and *recall* is the ratio of correct positive predictions of all of the total positives in the dataset. (Davis & Goadrich, 2006). High precision is favourable when there is high cost for false alarms, and high recall is beneficial if there is low cost for false alarms and all potential positive cases need to be identified. The formulas for Precision and recall are presented below (Davis & Goadrich, 2006):

$$Precision = \frac{True\ Positive}{True\ Positive+False\ Positive}, \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive+False\ Negative}, \quad (5)$$

PR curves show the relation between model's precision and recall at different classification thresholds. A classifier without discriminatory power would be a horizontal line proportional to the number of positive samples in the dataset. Similarly, to ROC curve, the PR AUC can be used as a metric in model evaluation (Tharwat, 2020).

3.3.3 Brier score

The Brier Score, also known as Mean Squared Error (MSE), is a measure proposed by Brier in 1950. Brier score is widely used as the objective function in machine learning algorithm. (Gong, 2021) It was selected to supplement ROC AUC metric as it describes the uncertainty of model's predictions and penalizes those predictions which are wrong but highly confident. Similarly, to ROC AUC metric, Bier score works well with imbalanced data and does not rely on threshold settings.

Brier score is the average deviation between the predicted default probability, and the realized default probability. The following formula was proposed by Brier (1950):

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t))^2, \quad (6)$$

Where f_t is the probability of the forecast at t instance, and o_t is the actual outcome at an t instance and N represents the number of forecasting instances. The value of the Brier score vary between 0 and 1. A perfect model would have a Brier score of 0 and a model which predicts all cases wrong would have a score of 1. (Gong, 2021)

3.4 Splitting the dataset into different time horizons

For the purpose of the analysis, the dataset is split to two datasets: One year before and two years before the event of default or non-default. This is done to test the modes' performance at two different time horizons and to only include the same information that would have been available at the time of making the prediction. The training and model selection based on the performance with test set is done on one year before dataset, as the purpose of the model is to predict default 12 months in advance. The predictive power of the model is also validated using the two years before dataset.

3.5 Training and testing sets

For model validation, the dataset should be divided into training set and test set. Using a separate dataset with data, that the model has not seen before for validation is critical to avoid overfitting. Both, the test set and the validation set should represent the variance in the sample. Generally, the training set should be 60-80% larger, but there are no rules for the set sizes. However, it should be considered that, the larger the holdout for the test set, the more information is left out from the training of the model. (Kohavi, 1995).

In this study the one year before dataset is split randomly into training and test set, where the training set contains 70% of the data and 30% is reserved for testing the model. The descriptive statistics of datasets are introduced in chapters 4.3.3 and 4.3.4.

3.6 Cross-Validation, dealing with unbalanced data and hyperparameter optimization

Due to limited amount of data, 10-fold cross-validation is used in the training set to evaluate the changes in the model with different folds, predictor variables and in optimizing the hyperparameters of each model to avoid overfitting. Cross-validation functions with dividing the dataset into set number of folds. The training of the model is done for each

fold and the changes in the model between the folds are evaluated to create an optimal model. Cross-validation is especially useful with limited sample size (Kohavi, 1995).

Hyperparameter optimization means optimizing models' parameters to improve the prediction performance of the model while avoiding overfitting. Different models have different parameters e.g. for decision trees, the number of learners and splits can be changed. (Hutter et al., 2019) The selected parameters and variables for each model is presented in chapter 5.

To deal with severe skew in the class distribution the training set folds are also balanced by random over-sampling of the minority class (defaulted). This is a commonly used method when dealing with rare events modelling such as fraud detection and default prediction. Imbalanced class distribution results in non-uniform misclassification costs when model is built and balancing the class distribution of the training dataset generally improves model's performance. (Branco et. al. 2016). Over-sampling was selected instead of under-sampling (removing part of the majority class) due to limited dataset. Over-sampling is done by randomly adding copies of the minority class to the dataset until the dataset is balanced. In some cases, this may result in overfitting especially with high over-sampling rates and result in a decrease in classifier performance (Chawla et al., 2002). Therefore, models were also created without over-sampling, and cases where model performance improved are reported separately.

Caret r-package selected for the Cross-validation, hyperparameter optimization and variable selection in this study.

4 Empirical Analysis

4.1 Case Company

The case company is a large retail company which operates in the Nordic countries and Baltics. The credit portfolio consists of products sold to the customers on credit for both, consumers and corporate customers with various payment terms.

Company operates in a low margin, high volume business with limited possibility for collaterals due to strict competition. Therefore, accurate and efficient credit risk assessment of the customers before delivering the products or granting credit limit plays a crucial role in ensuring profitability.

4.1.1 Credit risk management

Due to the low margin market the case company operates in, evaluating the credit worthiness of the commerce counterparties plays an important role in ensuring the profitability of the business. The company is not required to follow the Basel II/III regulations, however the payment institution authorisation issued by Finnish Financial Supervisory authority has its own requirements for the credit management such as the need for a specified non-business risk control function, the minimum required capital, the bad debt provision bookings according to International Financial Reporting Standard IFRS 9 and credit limits for the cards issued. (Financial supervisory authority 2005)

The company has own internal credit control team that is responsible for the credit risk assessment, collateral management and collection measures before sending the debt to a credit collection agency.

4.1.2 Current Credit Risk Management Process

Current process consists of three main sub processes: *Credit Scoring*, *Counterparty Monitoring* and *Collection*.

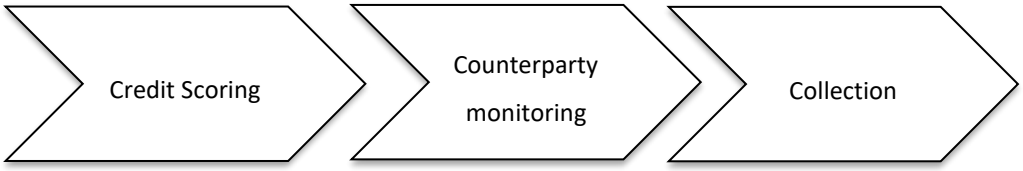

Figure 4. The main Credit Risk Management processes.

Credit Scoring takes place before the beginning of the customer relationship and in case of credit limit change requests. All the counterparties are assessed automatically before the first order, limit change, submitting tenders or approving the cash card application by a Credit Check –system that is fully integrated to the ERP-systems and external Credit Rating Agency. Based on the requested limit, internal payment behaviour information and information received from the Credit Rating Agency, the system then either approves the limit and gives a corresponding internal credit rating, declines the limit and informs the user the reason for decline or sends the query to Credit Control for manual assessment. Credit Check system's logic is currently a simplistic decision tree model that only approves or declines the request and relies heavily on the external data and the standard credit scoring done by the Credit Rating Agency. The model itself does not assess the probability of default and the rather simple decision tree has been created based on the expert opinion rather than statistical model.

Counterparty Monitoring is critical for successful credit risk management in the case company. Customers' external credit rating changes and payment defaults are monitored daily in order to take preventative measures to minimise EAD in case of a default. All counterparties are added automatically to the monitoring list after the first invoice is created. The Credit Rating Agency then sends daily all credit rating and payment behaviour changes to case company's database. The Credit Control adjusts the internal risk codes to appropriate level based on a manual assessment of these changes and takes other more extreme measures such as decreasing the limit, requesting for a collateral or prepayment to prevent possible defaulting.

Collection takes place in a case where customer has not paid the invoices by the due date. Dunning letters and remainder SMS messages are sent automatically to the customers after a manual review by credit controllers. If the debt is still open after the due date of the dunning letters, the debt is transferred to debt collection agency for further collection measures.

4.1.3 Challenges with current processes and systems

Despite the automated credit checking and constant monitoring, the vast number of customers makes managing the risk portfolio challenging and more automation is required to decrease the amount of manual risk assessment and to find the meaningful changes in customers' credit worthiness. The current rating-based monitoring requires a lot of manual work to assess whether the lowest ratings with highest probability of default requires immediate actions. Therefore, this study aims to find a model to predict PD more precisely and early enough to avoid credit losses.

As discussed in the targets of the study, the requirements for prediction model are *transparency*, *accuracy* and *cost effectiveness*. Managing the quality of data is important for any model. In the case company, the complexity of the system architecture makes data quality management of the data challenging. There are more than 60 separate systems integrated to two different ERPs. Therefore, the architecture is prone for data quality issues. To assess the correctness of the model and to explain the reasons behind credit decisions for customers, the data involved must be *transparent* for the end users. *Accuracy* is required so that the model adds value to new processes and gives minimal amount of false alarms. Cost plays always part when assessing the business case of every project and buying data from external sources is not free. Therefore, the model must be *cost effective* and require minimal amount of new data without compromising accuracy too much.

4.2 Currently used credit rating in risk assessment and monitoring

Company currently utilizes the credit rating provided by external credit rating agency Bisnode Oy. The external credit rating scoring model takes into account multiple internal and external data sources to conduct the rating score. The probability of default (PD) for different ratings is shown below in the Table 1.

Table 1. Statistical 12 Month PD and interpreted PD of the rating classes.

Rating	PD 12 Months	Interpreted PD
AAA	1,10 %	0 %
AA	1,40 %	0,5%
A	1,70 %	1 %
AN (New Company)	2,10 %	50 %
EI-R (No rating available)	1,60 %	50 %
B	17,40 %	70 %
C	39,30 %	100 %

As previously discussed, generally the rating model gives a good starting point for expert opinion, but challenges arise especially when assessing the risk of companies with rating B or C. As we can see, the probability of default is high for these ratings, but the case company would like to still grant credit for healthier B rated customers, and for C rated customers, there is a need to assess which of these need immediate risk mitigation actions. In some cases, customers with credit rating B or C could be profitable if the risk can be included in the pricing. Based on interviews with the credit control team, interpretation of the risk in practise is also tighter than the real statistical PD of the model. This is interpretation in the credit policy in practise is shown in Table 1. It is also noteworthy, that the expert opinion of the risk for A-AAA ratings are considered lower than what the official statistics show.

4.2.1 Setting the baseline: Credit rating's performance metrics

To set the comparable baseline for credit risk modelling, the ROC and AUC were calculated for both, the official 12-month PD and the interpreted PD with the both datasets. The results are presented in Table 2. The official 12-month PD outperformed the interpreted PD with in all the selected metrics with one year before and two years before datasets. The performance metrics of the external rating are shown in the table **Table 2** and the ROC and PR curves are presented in the Figure 6, Figure **5**Figure 8 Figure 7.

Table 2. The performance metrics of the external credit rating.

	One year before		Two years before	
Metric	12-month PD	Interpreted PD	12-month PD	Interpreted PD
ROC AUC	0,79	0,75	0,74	0,67
PR AUC	0,55	0,28	0,53	0,25
Brier score	0,17	0,23	0,11	0,24

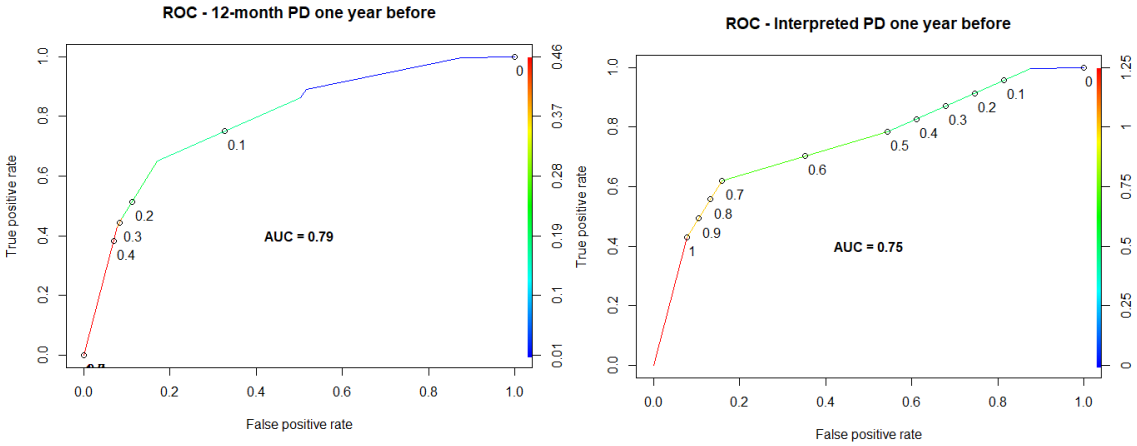

Figure 5. The ROC curves of the external credit rating – one year before dataset.

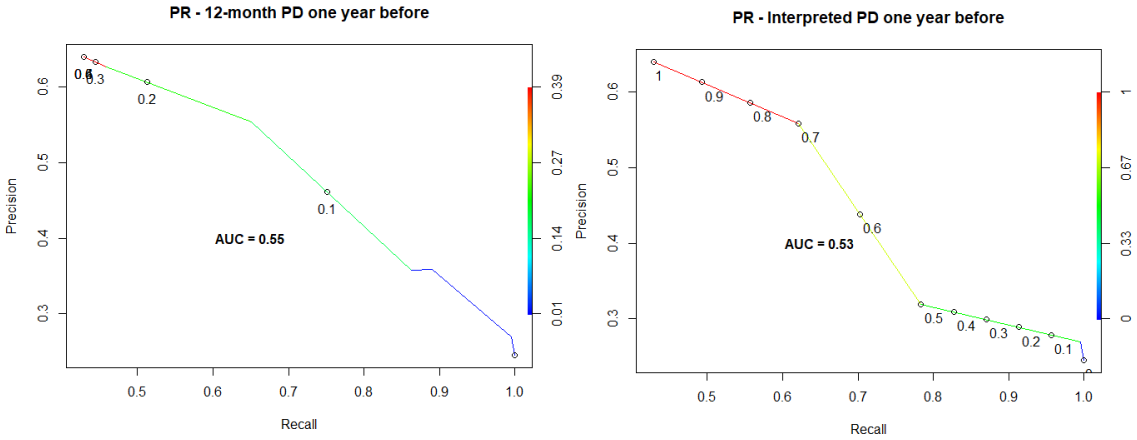

Figure 6. The PR curves of the external credit rating – one year before dataset.

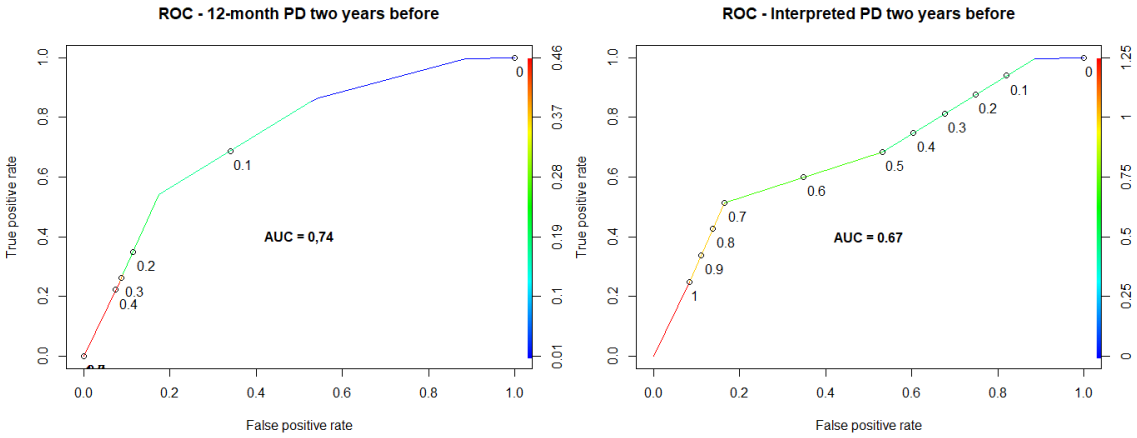

Figure 7. The ROC curves of the external credit rating – two years before dataset.

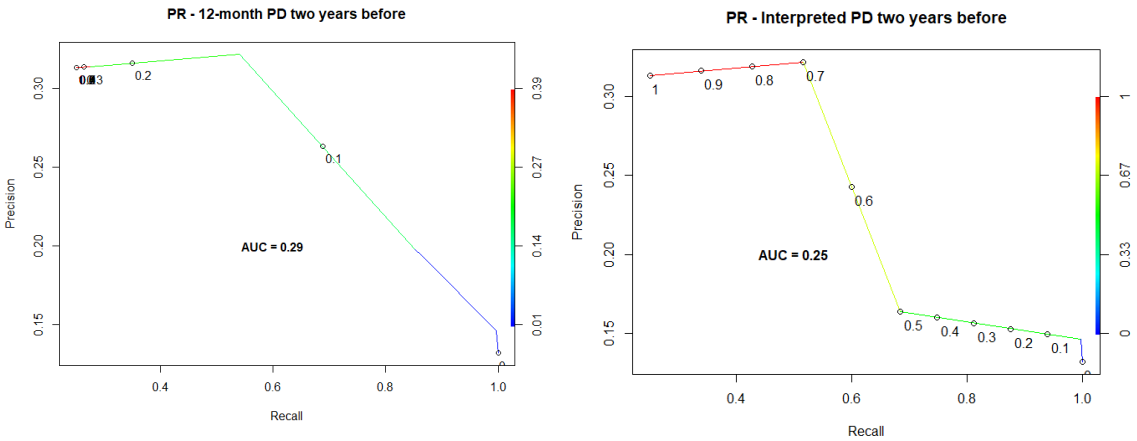

Figure 8. The PR curves of the external credit rating – two years before dataset.

4.3 Introduction of the Data

The data consists of the background information, historical financial key figures & ratios and payment behaviour data of 2471 Finnish companies from period 2009-2017. All external data is provided by external service provider Bisnode Finland Oy Dataset includes corporates from multiple different industries, however transportation and construction sectors represent 58% of the companies (Table 4). Most of the data consists of limited companies and other legal forms represent only 0,45% of the dataset (Table 3). These

are included in the dataset to see whether the credit code is applicable for other legal forms, however, the sample size is not high in enough to make final conclusions.

Table 3. Distribution of legal forms in the dataset.

Company type	2009	2010	2011	2012	2013	2014	2015	2016	2017	Grand Total
Limited Company	1	10	89	1150	1852	1967	2050	1966	694	2460
Sole Proprietorship				1	1	4	3			5
Limited Partnership			1	1	2	3	3	2	2	4
General Partnership							1	1		1
Cooperative				1	1	1	1	1		1
Grand Total	**1**	**10**	**90**	**1153**	**1856**	**1975**	**2058**	**1970**	**696**	**2471**

Table 4. Distribution of industries in the dataset.

Industry	Grand Total
Land transport and transport via pipelines	829
Construction of buildings	601
Manufacture of food products	253
Wholesale and retail trade and repair of motor vehicles and motorcycles	230
Rental and leasing activities	140
Crop and animal production, hunting and related service activities	133
Legal and accounting activities	62
Real estate activities	46
Water collection, treatment and supply	44
Mining of coal and lignite	33
Electricity, gas, steam and air conditioning supply	19
Human health activities	15
Accommodation	13
Publishing activities	12
Education	11
Creative, arts and entertainment activities	11
Activities of membership organisations	11
Financial service activities, except insurance and pension funding	8
Grand Total	**2471**

It should be also noted that all the customers have passed the initial credit assessment when starting the customer relationship, so the company sample is not randomly selected and does not therefore necessarily represent a conclusive population of all companies in Finland. This limitation was accepted, because the purpose of this study is to combine internal payment behaviour data with external credit rating and confirm the hypothesis that this may improve the model performance.

The frequency of the financial statement observations is presented in Table 5. It should be noted that majority of the data in the one year before dataset is from year 2016 and in the two years before dataset from year 2015. The observations that are from more than two years before the event of default or non-default were excluded from this study.

Table 5. Frequency of Financial Statement observations.

Dataset	2009	2010	2011	2012	2013	2014	2015	2016	2017	Grand Total
one year before			2	68	174	202	245	1970	696	2398
two years before		1	22	61	64	98	1813			2059
>two years before	1	9	66	1024	1618	1676				1816
Grand Total	**1**	**10**	**90**	**1153**	**1856**	**1975**	**2058**	**1970**	**696**	**2471**

4.3.1 Definition of Default in the Study

As discussed in chapter 3.1 there is still no consensus in the academic research on the definition for defaulting. By reviewing the historical data of the case company's corporate customers' recovery rates from period 2014-2017, we can conduct, that the recovery rate of receivables that are over 90 days past due varies between 17% and 25 % with the average of 17,3 %. As there are no products with more than 17% margin in the portfolio, it can be concluded that the recovered funds do not cover the costs of the products and losses are evident.

Therefore, the chosen definition for default in this sample is a defined by bad debt bookings. The bookings in the case company are made in compliance with the Basel II (BSBS,2006) accord: Either the obligor is unlikely to pay its credit obligations in full, for example in case of a Bankruptcy or debt restructuring, or the receivables are past due more than 90 days.

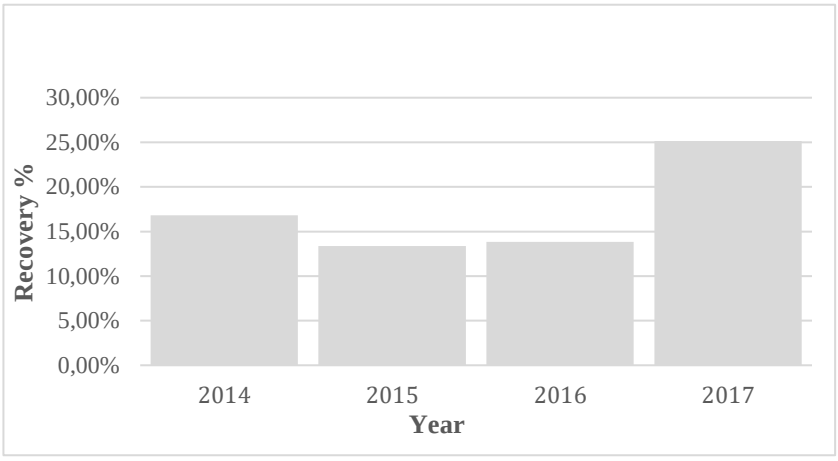

Figure 9. More than 90 days overdue recovery rate.

With the definition set above, the dataset is unbalanced and only 22,6% of the sample companies defaulted during the research period. The distribution of default presented below in Table 6 also shows that majority of the defaults in the dataset occurred in years 2015-2016.

Table 6. Distribution of default and non-default companies.

Defaulted	2013	2014	2015	2016	2017	2018	Grand Total
No						1913	1913
Yes	5	111	152	153	137		558
Grand Total	**5**	**111**	**152**	**153**	**137**	**1913**	**2471**

4.3.2 Variable selection

The variables chosen in this study, consists of information of the company type, industry, internal and external payment behaviour as well as key figures describing the profitability and solvency of the companies from financial statements, including both, absolute figures as well as financial ratios.

It was decided to include all currently available data and rather than excluding any variables based on the statistical significance in previous studies. The variable selection is done for each model separately to create best performing model while avoiding overfitting to see if different models can find different underlying relationships between the predictor variables.

4.3.2.1 Characteristics and Payment Behaviour

The selected characteristic and payment behaviour predictors are presented in Table 7. Industry, legal form and company's age in months originate from the Finnish Trade register information. Industry is based on company's registered NACE code level 1 according to the industry standard classification system used in the European Union. Rating represents the external rating of the company and paydex index represents the internal payment behaviour. The default information originates from the Finnish credit information register. Default to net sales ratio was calculated to proportion the default amount to the size of the company.

Table 7. Characteristics and payment behaviour predictors.

Category	Predictor variable	Description
Characteristics	Default status	Information whether the company has defaulted
	Industry	Industry based on NACE code
	Legalform	Legal form of the company
	Rating	External credit rating
	Age in months	Age of the company in months
Payment Behaviour	Average paydex	Payment behaviour index: Paydex = 80 payments on time Paydex > 80 payments before due date Paydex < 80 payments after due date
	Average amount of defaults	Average amount of public payment defaults in euros
	Default to net sales ratio	$\frac{\text{Average amount of defaults}}{\text{Turnover}}$
	Number of defaults	The number of cumulative defaults at given time

4.3.2.2 Financial key figures

The Financial key figures presented in Table 8 originate from companies' financial statements and represent the activity, profitability and solvency of the company. Unlike the financial ratios, which are comparable with companies of different sizes, these are raw numbers, which in practice, also describe the size of the company. These were included in the dataset to see whether some models would benefit from this raw data.

Table 8. Financial Key Figures.

Category		Predictor variable	Description
Financial key figures	Activity & size	Change of turnover	Turnover change from previous financial statement
		Change of turnover precentage	Same as above in %
		Employee costs 12Months	Employee costs in the last financial statement
		Turnover	Turnover in the financial statement
	Profitability	Operating result	Earnings before interest and taxes
		Net result	Operating result +/- interest +/- taxes excl. Extraordinary items
		Financing result	Net result +/- depreciation & armorization
		Fiscal year result	Net P/L for the year
	Solvency	Equity	Shareholders equity

4.3.2.3 Financial ratios

The financial ratios presented in Table 9 are derived from the financial statements. Some of the ratios have been directly stored to case company's database without the fundamental financial key figures needed for calculation. As discussed in chapter 2, the financial ratios have been successfully used in the previous studies. One benefit of using ratios is that they make the figures comparable between companies of different size, and show additional information of company's profitability, solvency and liquidity.

Table 9. Financial ratios.

Category		Predictor variable	Description
Financial ratio	Profitability	Fiscal result to turnover	$\frac{\text{Fiscal result}}{\text{Turnover}}$
		Net result to turnover	$\frac{\text{Net result}}{\text{Turnover}}$
		Operational result to turnover	$\frac{Operational\ \boldsymbol{result}}{\text{Turnover}}$
		Return on equity	$\frac{\text{Net result}}{\text{Equity}} \times 100$
		Return on investments	$\frac{\text{Operating result}}{(\text{Equity} - \text{interest bearing debt})} \times 100$
		Return on total assets	$\frac{\text{Net result}}{\text{Average Assets}} \times 100$
	Solvency & Liquidity	Debt to net sales ratio	$\frac{\text{Total debt}}{\text{Turnover}} \times 100$
		Equity ratio	$\frac{\text{Equity}}{\text{Total assets}} \times 100$
		Gearing	$\frac{\text{Total debt}}{\text{Equity}} \times 100$
		Quick ratio	$\frac{\text{Current assets} - \text{Inventory}}{\text{Current Liabilities}}$
		Current ratio	$\frac{\text{Current assets}}{\text{Current Liabilities}}$
		Days of payables outstanding	$\frac{\text{Average Accounts Payable}}{\text{Costs of Goods sold}}$
		Days of receivable outstanding	$\frac{\text{Average Accounts Receivable}}{\text{Turnover}}$

4.3.3 Descriptive statistics

The descriptive statistics for the predictor variables are shown in tables Table **11**, Table **10** and Table **12**. It can be clearly seen that the defaulted companies have negative mean profitability metrics, highter payment delays, more public payment defaults and higher gearing than the companies that did not default.

Table 11. The descriptive statistics of the dataset of the full dataset.

Descriptive statistics all companies

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	5,776	76.3	6.0	11.5	99.0
Number_of_defaults	9,815	0.1	0.9	0	46
Average_amount_of_defaults	9,815	181.7	3,101.1	0.0	162,201.0
Turnover	9,815	7,764.8	39,980.6	−44	900,402
Change_of_turnover	9,815	149.1	12,906.4	−992,608	217,911
Employee_costs_12Months	8,688	1,615.0	10,194.0	−19	330,256
Operating_result	9,019	486.1	6,055.1	−28,582	265,721
Net_result	9,026	347.9	5,405.4	−95,536	223,329
Financing_result	9,024	657.9	6,427.1	−95,536	253,478
Fiscal_year_result	9,023	250.4	5,496.0	−95,536	240,588
Return_on_total_assets	9,812	59.2	304.2	−8,429	3,261
Return_on_investments	9,238	111.3	929.5	−15,725	76,985
Return_on_equity	9,215	1.6	838.4	−32,100.0	37,300.0
Equity	9,016	2,442.6	18,487.0	−38,135	706,515
Equity_ratio	9,815	226.6	467.1	−9,105	999
Debt_to_net_sales_ratio	8,578	51.5	3,044.4	−263,777.3	75,044.4
Gearing	9,215	20.6	442.4	−4,500.0	24,800.0
Quick_ratio	9,815	4.1	2.9	−0.8	9.9
Current_ratio	9,815	1.2	1.2	−0.3	9.8
Days_of_receivable_outstanding	8,804	42.8	562.9	−121	52,564
Days_of_payables_outstanding	8,511	428.2	11,125.1	−24,455	782,925
Age_in_months	9,815	195.0	166.4	0	1,400
Default_to_net_sales_ratio	9,815	0.2	6.9	0.0	605.3
Operational_result_to_turnover	9,049	0.04	0.8	−54.0	38.5
Net_result_to_turnover	9,055	0.1	8.8	−54.0	830.0
Fiscal_result_to_turnover	9,052	0.1	8.9	−56.0	838.9
Change_of_turnover_precentage	9,815	−163.6	11,775.0	−992,608.0	315.5

Table 10. The descriptive statistics of the defaulted companies.

Descriptive statistics defaulted companies

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	626	70.0	10.0	11.5	87.9
Number_of_defaults	1,346	0.3	1.9	0	38
Average_amount_of_defaults	1,346	393.7	3,619.0	0.0	84,134.0
Turnover	1,346	2,167.7	11,749.0	−6	320,829
Change_of_turnover	1,346	−1,036.4	32,148.4	−992,608	85,025
Employee_costs_12Months	510	453.7	1,173.8	−19	9,759
Operating_result	550	−24.4	544.6	−6,668	3,287
Net_result	557	−329.9	4,650.0	−95,536	3,288
Financing_result	555	−259.0	4,654.2	−95,536	3,938
Fiscal_year_result	554	−335.5	4,658.7	−95,536	3,275
Return_on_total_assets	1,346	−101.7	616.9	−7,875	1,946
Return_on_investments	772	15.4	2,955.3	−15,725	76,985
Return_on_equity	770	−22.4	593.3	−11,400.0	3,400.0
Equity	549	273.9	1,230.7	−3,546	14,396
Equity_ratio	1,346	−127.7	840.9	−9,105	988
Debt_to_net_sales_ratio	526	68.2	109.8	−133.3	1,036.4
Gearing	770	160.1	1,347.3	−4,500.0	24,800.0
Quick_ratio	1,346	4.0	2.9	−0.8	9.9
Current_ratio	1,346	0.5	0.7	−0.3	6.7
Days_of_receivable_outstanding	752	29.9	43.4	−121	456
Days_of_payables_outstanding	732	372.1	3,709.6	0	78,110
Age_in_months	1,346	140.3	131.9	0	1,144
Default_to_net_sales_ratio	1,346	0.9	16.8	0.0	605.3
Operational_result_to_turnover	580	−0.001	0.4	−1.4	8.3
Net_result_to_turnover	586	−0.03	0.4	−1.6	8.3
Fiscal_result_to_turnover	583	−0.03	0.4	−1.6	8.3
Change_of_turnover_precentage	1,346	−1,192.8	31,787.8	−992,608.0	315.5

Table 12. The descriptive statistics of the not defaulted companies.

Descriptive statistics not defaulted companies

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	5,150	77.1	4.7	16.5	99.0
Number_of_defaults	8,469	0.03	0.6	0	46
Average_amount_of_defaults	8,469	148.0	3,009.5	0.0	162,201.0
Turnover	8,469	8,654.4	42,718.0	−44	900,402
Change_of_turnover	8,469	337.5	5,351.3	−106,671	217,911
Employee_costs_12Months	8,178	1,687.4	10,498.7	0	330,256
Operating_result	8,469	519.3	6,245.7	−28,582	265,721
Net_result	8,469	392.5	5,448.7	−48,535	223,329
Financing_result	8,469	718.0	6,522.2	−38,999	253,478
Fiscal_year_result	8,469	288.8	5,544.4	−48,530	240,588
Return_on_total_assets	8,466	84.8	204.9	−8,429	3,261
Return_on_investments	8,466	120.1	382.5	−13,643	17,928
Return_on_equity	8,445	3.8	857.3	−32,100.0	37,300.0
Equity	8,467	2,583.2	19,065.9	−38,135	706,515
Equity_ratio	8,469	283.0	342.6	−7,828	999
Debt_to_net_sales_ratio	8,052	50.4	3,142.2	−263,777.3	75,044.4
Gearing	8,445	7.9	215.3	−2,141.0	18,836.0
Quick_ratio	8,469	4.1	2.9	−0.2	9.9
Current_ratio	8,469	1.3	1.2	0.0	9.8
Days_of_receivable_outstanding	8,052	44.0	588.4	0	52,564
Days_of_payables_outstanding	7,779	433.5	11,581.1	−24,455	782,925
Age_in_months	8,469	203.8	169.6	0	1,400
Default_to_net_sales_ratio	8,469	0.1	3.2	0.0	229.3
Operational_result_to_turnover	8,469	0.04	0.8	−54.0	38.5
Net_result_to_turnover	8,469	0.1	9.1	−54.0	830.0
Fiscal_result_to_turnover	8,469	0.1	9.2	−56.0	838.9
Change_of_turnover_precentage	8,469	−0.04	5.2	−451.0	1.0

4.3.4 Correlation matrix

The correlation matrix presented in Figure 10 was built with using corrplot r-package for the full dataset. The pairwise Pearson's correlation coefficient correlation is displayed by the colour gradient and the crosses represent the absence of significance (p-value > 0,05). As expected, all profitability key figures and ratios are highly correlated. Also, average paydex, equity ratio and age in months have a significant negative correlation with the default status (company defaulted or did not default). The correlations are similar with the one year before and two years before datasets presented in Figure 11 and Figure 12.

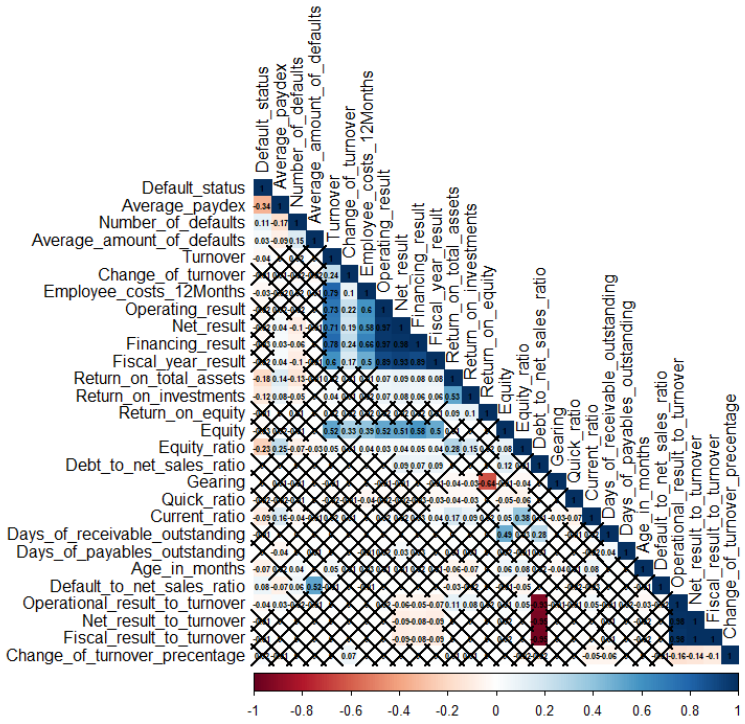

Figure 10. The correlation matrix – full dataset.

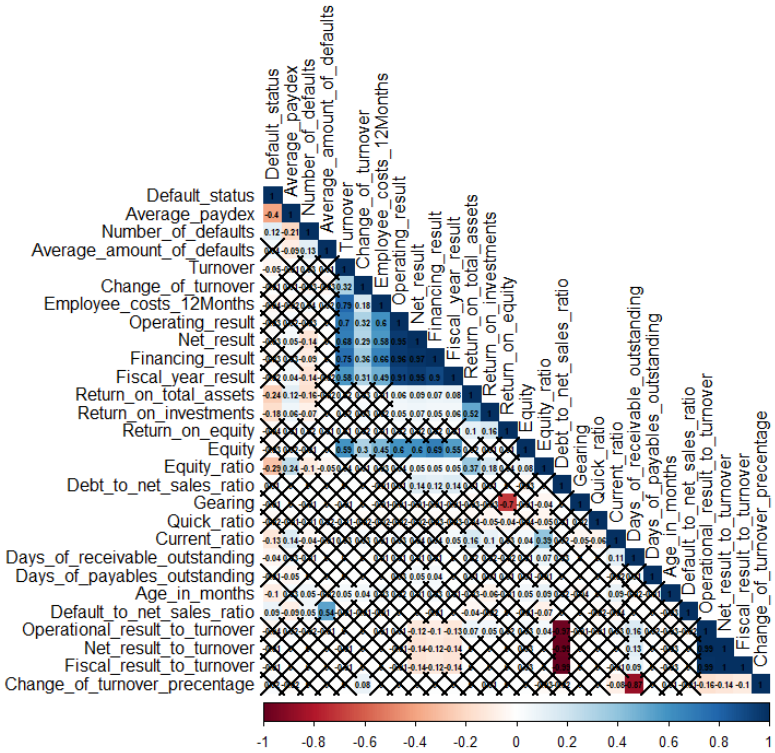

Figure 11. The correlation matrix – one year before dataset.

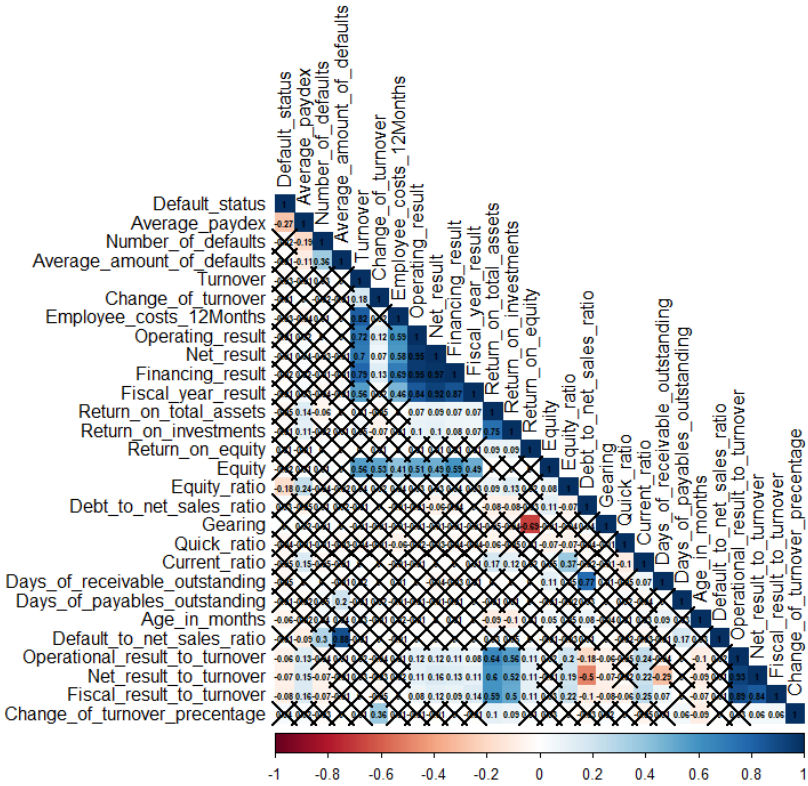

Figure 12. The correlation matrix – two years before dataset.

4.4 Missing data and imputation

One key finding is that the dataset has missing values. In fact, 4,9% (Figure 13) of the data is missing. This is of relatively common in and can have a noteworthy impact on the conclusions that can be drawn from the data (Graham, 2009). One of the requirements for the model is the ability to handle missing data. In this chapter a short summary of methods is presented.

Most statistical analyses require complete dataset and software packages assume that all variables in the model have been measured and automatically delete cases with missing data. Wilkinson (1999) concluded that deletion methods are among the worst methods to deal with missing data and may result in severe loss of statistical power. In addition, in the case of the case company the likelihood of missing data is large, as financial statements are usually published only once per year and with different requirements of

the included information, depending on the company form and size. One key requirement for the model is to have up to date defaulting probability for all companies, even if some of the data is missing. Therefore, the proposed solution must include a method to deal with the missing data.

A wide variety of techniques to deal with missing data was used until Rubin's (1976) journal set the framework for different types of missing data and highlighted the importance of considering the process that causes the missing data and it's impacts on analyses. Most traditional techniques to deal with the issue include deletion, use of mean and other single imputation approaches (Peugh & Enders, 2004). The more modern methods, such as maximum likelihood estimation and multiple imputation, are considered "state of the art" techniques and often provide better results (Schafer & Graham 2002).

First step to choose the method is to evaluate understand the type of missing data and possible correlations. Rubin (1976) classified the into three categories: missing completely at random (MCAR); missing at random (MAR); and missing not at random (MNAR). Categorisation is done based on the relationship between variables and the likelihood of missing data. Each type of missing data dictates the performance of imputation techniques (Little et al., 2012; Sterne et al., 2009; Dziura et al., 2013).

4.4.1 Types of missingness

Missingness can be defined as missing completely at random (MCAR) when there is no systematic reason for missing observation. In other words, the relationship between either observed or unobserved (missing) data and missingness. (Schafer & Graham, 2002)

Missing data is considered as missing at random (MAR) when the missingness is random and may depend on the observed variables, but not on unobserved (missing) data. (Schafer & Graham, 2002)

If the m definitions of MCAR or MAR are not met, the missingness is considered as missing not at random (MNAR). In this case, the probability of missingness depends on both, the observed data, and the unobserved (missing) data. (Schafer & Graham, 2002)

In summary, in case of MCAR, the missing data can be ignored (deleted) or imputed to preserve sample size as missing data is considered completely random, in case of MAR different imputation methods can be used and in case of MNAR the traditional imputation methods cannot be used as missing data is dependent of the unobserved data.

While validating the accuracy of the imputation methods is out of the scope of this study, it is important to understand whether imputation is feasible. Generally, the multiple imputation methods require that the data is MAR, and the missing values can be replaced by predictions derived by the observable portion of the dataset (Little & Rubin, 2019).

4.4.2 Missingness in the dataset

To get a better understanding of the missing data and type of the missing data, the locations and distribution of missing values across all variables are visualised in Figure 13.

Figure 13. Missing data in the dataset.

Based on this it can be conducted, that the majority of the missing data comes from variable *average amount of paydex* with 41,15% of the data missing followed by *Rating* variable, which is missing with 24,07% of observations. It is also clear that the missing data is not MCAR as also certain financial statement KPIs are clustered i.e. if one KPI is missing, likely other data is missing for the same company as well.

Figure 14 illustrates the presence of missing values related with missing data in other variables. This verifies the previous hypothesis that there is a relationship between multiple missing variables, especially between previously mentioned *rating* and *average amount of paydex.*

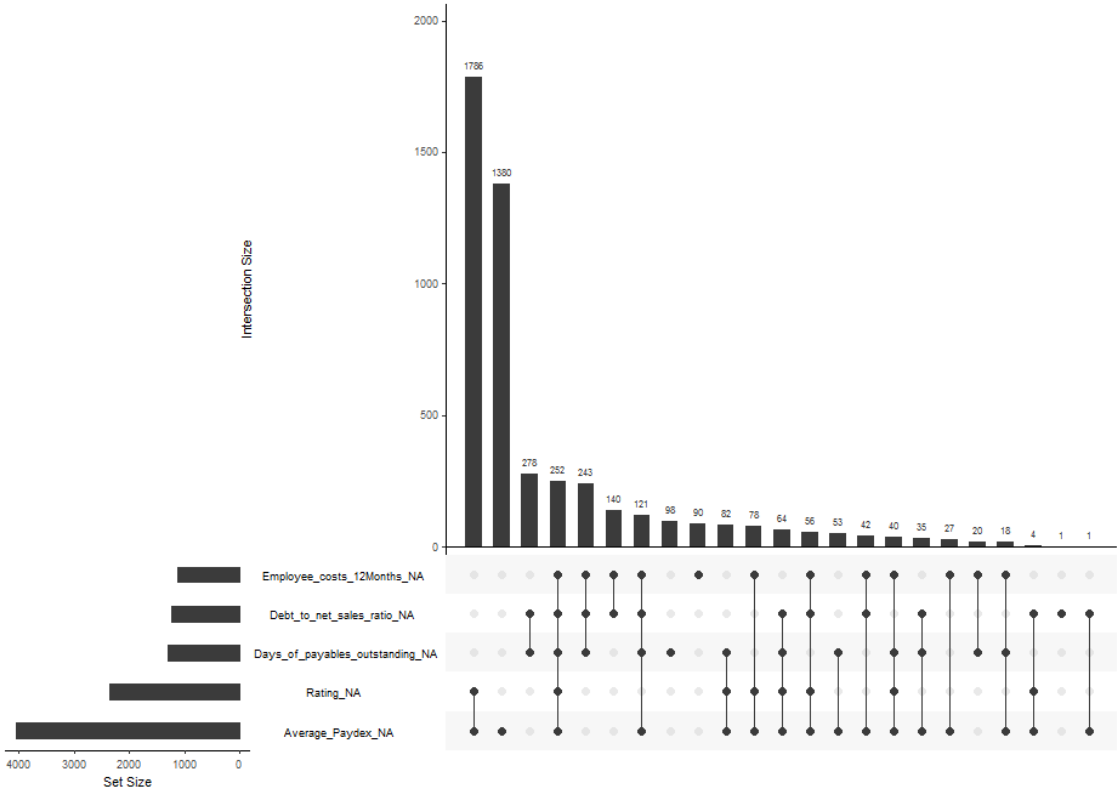

Figure 14. Relation of the missing values in the full dataset.

The amount of missing values in *rating* and *average amount of paydex* is slightly worrying as the statistical significance in determining the probability of default is high. However, the situation is much better with the one year before dataset. Figure 15 and Figure 16 illustrate that the amount of missing values in *rating* is reduced to 9,5% and in *average amount of paydex* to 15,2% one year before defaulting, while the total amount

of missing values increases slightly to 6,3%. Increase comes from missing financial statements for defaulted companies, which may be result of not publishing the financial statements on time to hide the challenges in business performance. Although it is compulsory to file financial statements to the Finnish patent and registration office (PRH), there aren't any consequences for filing the financial statement late or not filing it at all.

Figure 15. Missing data one year before dataset.

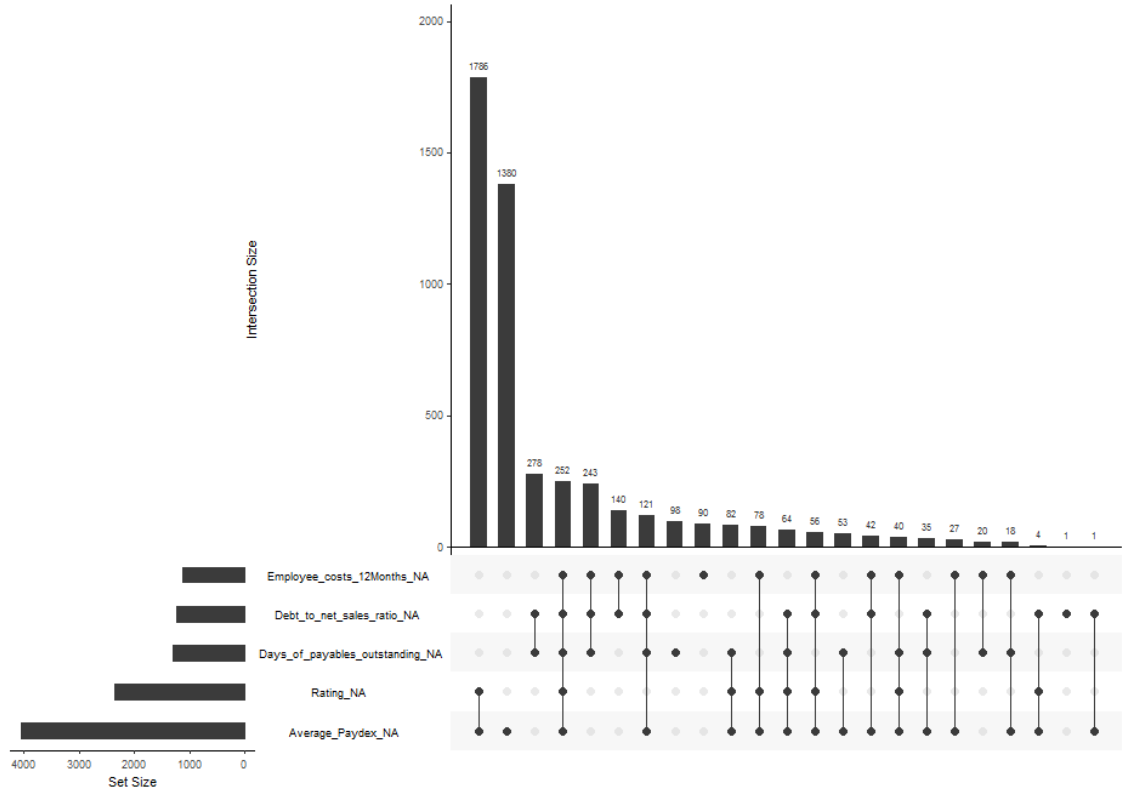

Figure 16. Relation of the missing values one year before dataset.

It should be noted that all the missing data is originally derived from the same source. Whether the reason for this lies in in the case company's data warehouse, in the data transfer process, or in the service providers processes, needs to be investigated further, but is out of the scope of this case study. However, these findings suggest that the missing values are not MCAR, but also not MNAR, as the probability that a variable value missing is not related to missing data values, when the missing data seems to derive from process issue. In addition, this finding emphasizes the need for an imputation process in the model, as missing data seems to be common challenge for the case company.

4.4.3 Imputation method

Comparison of various imputation methods is out the scope of this study, so the selection of imputation method is done based on previous research.

Some commonly used multiple imputation methods include k-nearest neighbours, multivariate imputation by chained equations (MICE), Pattern Alternating Maximization Algorithm (MissPALasso) and random forest (MissForest) algorithms. Several studies have

concluded that MissForest outperforms other imputation algorithms, and it has been used as a benchmark for other unsupervised imputation methods (Stekhoven & Buhlmann, 2012; Waljee et al., 2013; Shah et al., 2014; Tang & Ishwaran, 2017; Ramosaj & Pauly, 2019). In addition, MissForest algorithm works with both, continuous and categorical variables, is available in R and Python, does not require tuning and is therefore a good candidate for automating the imputation process when the model is implemented to case company's processes.

4.4.4 Imputation result

MissForest imputation is done in this study by utilizing missForest r-package, which utilizes RF proximity algorithm proposed by Breiman. The missForest provides an out-of-bag error (OOB) imputation error estimate for categorical and continuous variables. The first value is the normalized root mean squared error NRMSE, for the continuous variables in the imputed dataset. The second value is the proportion of falsely classified entries (PFC) in the imputed categorical variables. In both cases value close to 0 is considered of a good performance and a value around 1 is considered as a bad performance. (Oba et al., 2003)

The imputation resulted in NRMS of 0,34 for numeric variables and PFC of 0,13 for the categorical variables. Hence, we can conclude that while the performance was not perfect for the numeric variables, for the categorical variables the model performed well.

4.4.5 Descriptive statistics of the imputed dataset

The descriptive statistics and the correlation matrix are presented below in and Table 18. As expected, the imputation did slightly increase the pairwise correlation shown in the correlation matrix. The descriptive statistics are similar to the original dataset. The descriptive statistics for the imputed one year before and two years before datasets for defaulted and not defaulted companies are shown in appendices chapter 9.1.

Figure 18. Descriptive statistics of the full imputed dataset.

Descriptive statistics all companies

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	9,815	76.3	4.9	11.5	99.0
Number_of_defaults	9,815	0.1	0.9	0	46
Average_amount_of_defaults	9,815	181.7	3,101.1	0.0	162,201.0
Turnover	9,815	7,764.8	39,980.6	−44	900,402
Change_of_turnover	9,815	149.1	12,906.4	−992,608	217,911
Employee_costs_12Months	9,815	1,634.2	9,830.8	−19.0	330,256.0
Operating_result	9,815	420.3	5,816.3	−28,582.0	265,721.0
Net_result	9,815	258.0	5,207.8	−95,536.0	223,329.0
Financing_result	9,815	594.2	6,176.7	−95,536.0	253,478.0
Fiscal_year_result	9,815	165.2	5,293.2	−95,536.0	240,588.0
Return_on_total_assets	9,815	59.2	304.1	−8,429.0	3,261.0
Return_on_investments	9,815	71.8	945.0	−15,725.0	76,985.0
Return_on_equity	9,815	−19.9	877.0	−32,100.0	37,300.0
Equity	9,815	2,372.3	17,735.2	−38,135.0	706,515.0
Equity_ratio	9,815	226.6	467.1	−9,105	999
Debt_to_net_sales_ratio	9,815	215.0	3,048.9	−263,777.3	75,044.4
Gearing	9,815	48.6	472.1	−4,500.0	24,800.0
Quick_ratio	9,815	4.1	2.9	−0.8	9.9
Current_ratio	9,815	1.2	1.2	−0.3	9.8
Days_of_receivable_outstanding	9,815	82.9	582.8	−121.0	52,564.0
Days_of_payables_outstanding	9,815	2,794.5	16,231.7	−24,455.0	782,925.0
Age_in_months	9,815	195.0	166.4	0	1,400
Default_to_net_sales_ratio	9,815	0.2	6.9	0.0	605.3
Operational_result_to_turnover	9,815	−2.0	120.9	−9,486.7	314.1
Net_result_to_turnover	9,815	−4.1	254.3	−19,837.7	830.0
Fiscal_result_to_turnover	9,815	−4.4	270.7	−19,405.9	838.9
Change_of_turnover_precentage	9,815	−163.6	11,775.0	−992,608.0	315.5

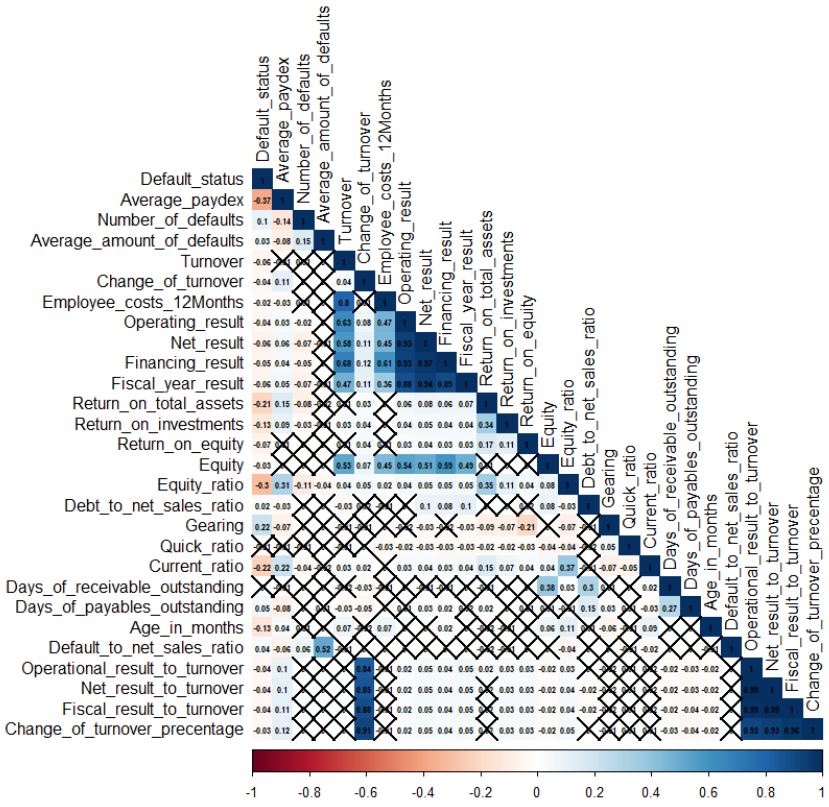

Figure 17. Correlation matrix of the full imputed dataset.

4.5 Splitting data into training and test set

The imputed data was split to training and test set using r-package Caret's createDataPartition function, which creates a stratified random split of the dataset. Stratified random split means, that the dataset is split into train and test sets in a way that the proportions of each class from the original dataset is preserved for each split.

70% of the dataset is reserved for training of the model and 30% as a hold-out sample to test the models with unforeseen data.

5 Model Development and Performance

In this chapter, the development of the models and the performance results with the training data and testing data are covered. All models were built using R version 4.1.2. with integrated development environment Rstudio.

All models were 10-fold cross validated with stratified folds. Minority class random up-sampling was used for all other models except for the selected Neural Network model, which performed significantly better without the sampling. Variable scaling was used for all other than tree-based algorithms, namely CART, Adaboost and Random Forest. These algorithms utilize rules and do not require normalization. Finally, each model's performance metrics are compared with the each other and the baseline set by the external rating.

Variable selection was done with the available feature selection functionalities for each model. The simpler models were preferred as they are generally easier to interpret and can therefore be considered better (Gosiewska et. al., 2020). The relative variable importance is shown for each model using command `variableimp` from the CARET r-package. The command uses different methods for each model type in showing the relative importance of each predictor.

5.1 Penalized Multivariate Discriminant Analysis (MDA)

Penalized MDA model was built using r-package MDA with method pda2. The pda2 model was optimized by selecting the optimal model with using the largest ROC. The best model was achieved with degrees of freedom (df) value of 6.

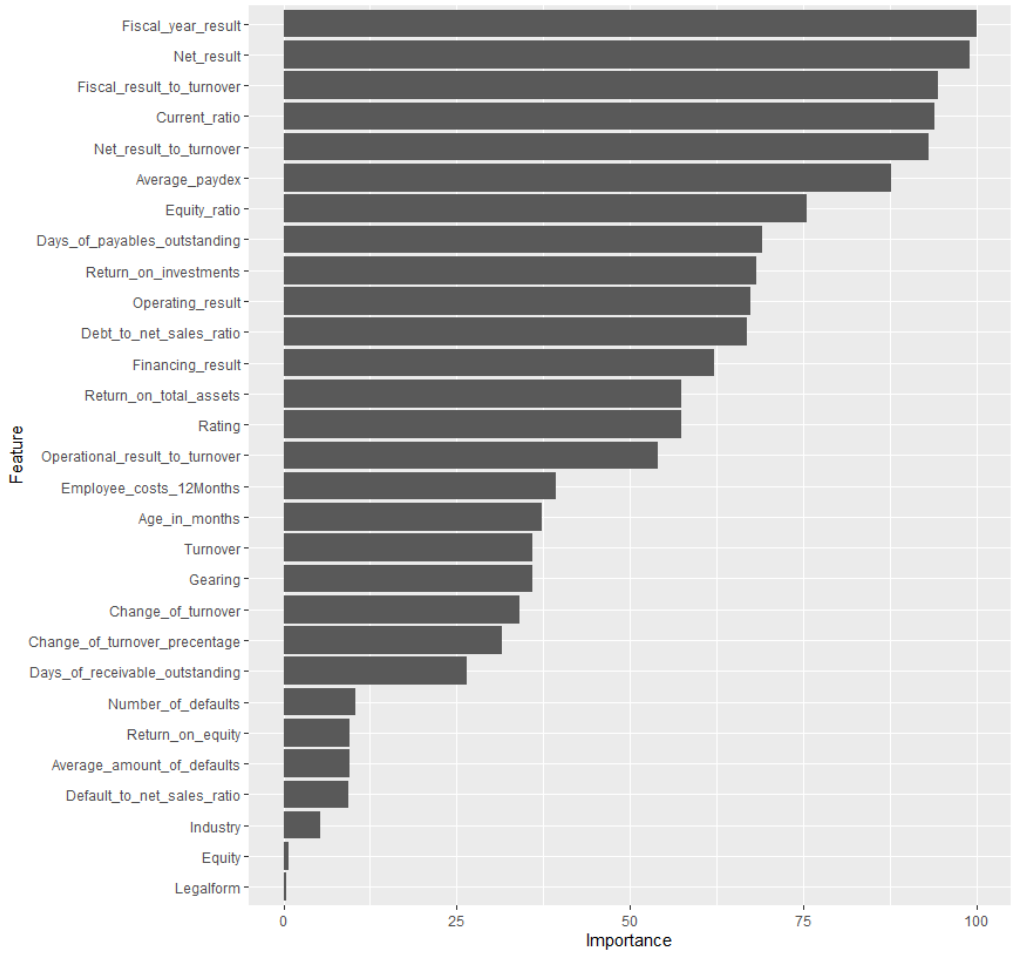

Figure 19. The first MDA model, the importance of the predictors.

The resulting model can be considered quite complex with 29 predictors presented in the Figure 19. The first MDA model, the importance of the predictors It should be noted that all variables describing company's result have with high correlation and were included in the model.

The resulted metrics in Table 13 show, that the model performed better than the baseline.

Table 13. The performance metrics of first MDA model.

Metric	One year before	Two years before
ROC AUC	0,91	0,85
PR AUC	0,78	0,51
Brier score	0,19	0,19

To create a simpler model, all other profitability predictors but fiscal result and variables legal form and equity, which have low relative importance in the model, were removed. The simpler model with predictors shown in Figure 20 performed similarly to the more complex model, and PR AUC and Brier score even improved (Table 14). This was selected as the final model.

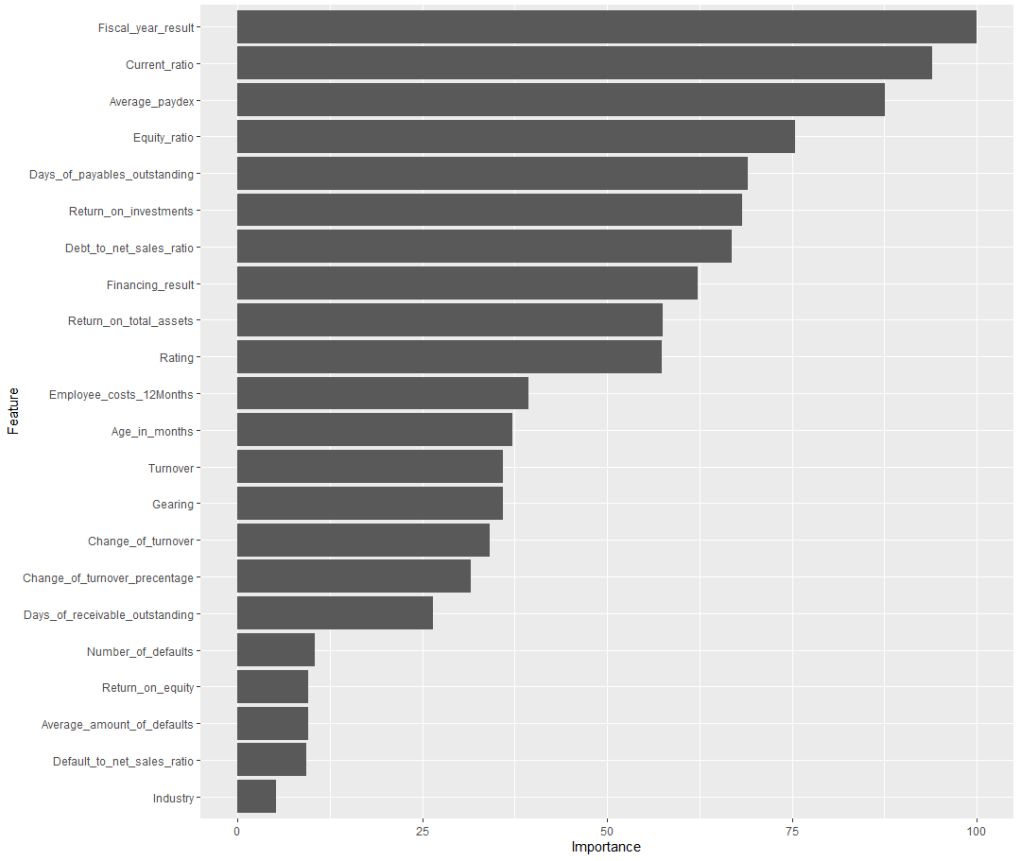

Figure 20. The predictor importance in the final MDA model.

Table 14. The performance metrics of final MDA model.

Metric	One year before	Two years before
ROC AUC	0,91	0,85
PR AUC	0,79	0,52
Brier score	0,18	0,19

The ROC and PR curves are presented in Figure 22 and Figure 21. It should be noted, that with higher thresholds, the Precision decreases significantly in the PR curve for the one year before dataset.

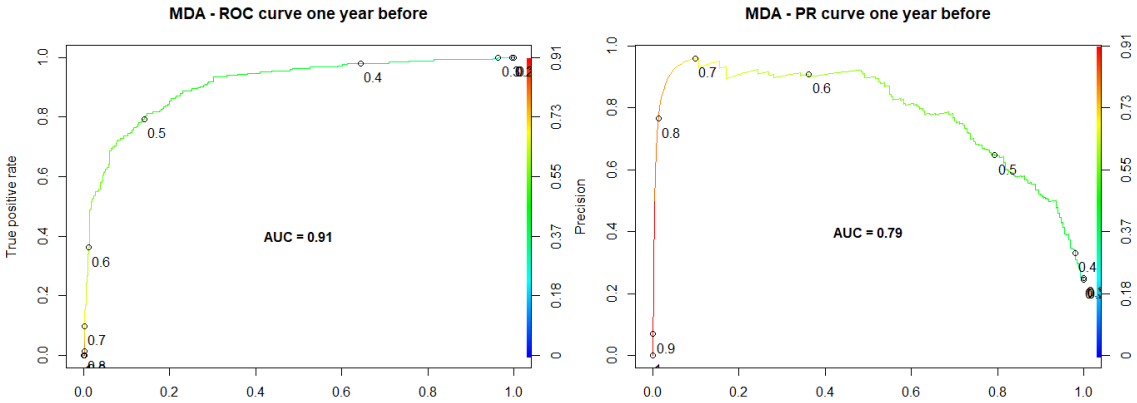

Figure 22. MDA model's ROC and PR curves – two years before dataset.

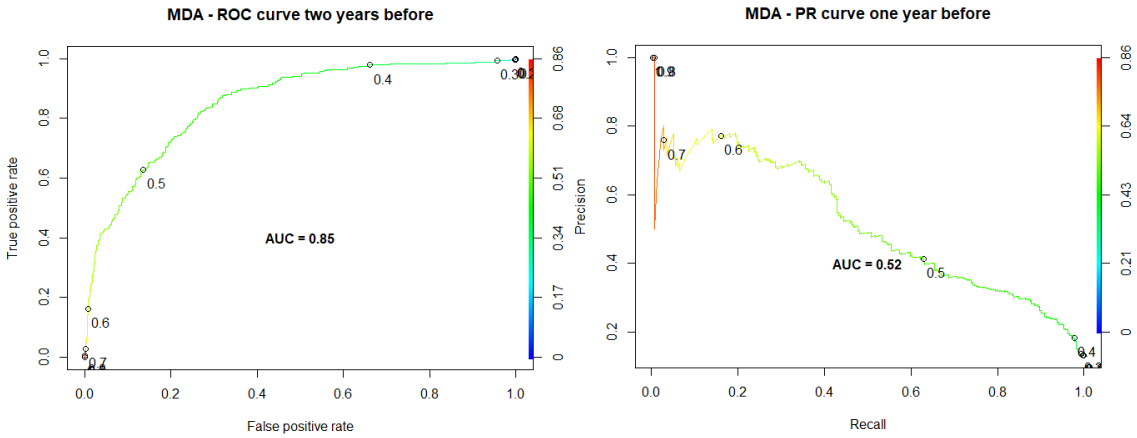

Figure 21. MDA model's ROC and PR curves – one year before dataset.

5.2 Logistic Regression

Two logistic regression models were created by using two variable selection methods.

5.2.1 Stepwise Logistic Regression

Stepwise logistic regression was built with MASS r-package's glmStepAIC-method. The stepwise selection direction was set to backward, which means that the variable selection begins with a model that contains all variables and then the least significant variables are removed step by step.

The variable selection according to importance is presented in. The stepwise variable selection resulted in quite complex model with 30 predictors included in the final model presented in Figure 23 along with the relative variable importance in the model.

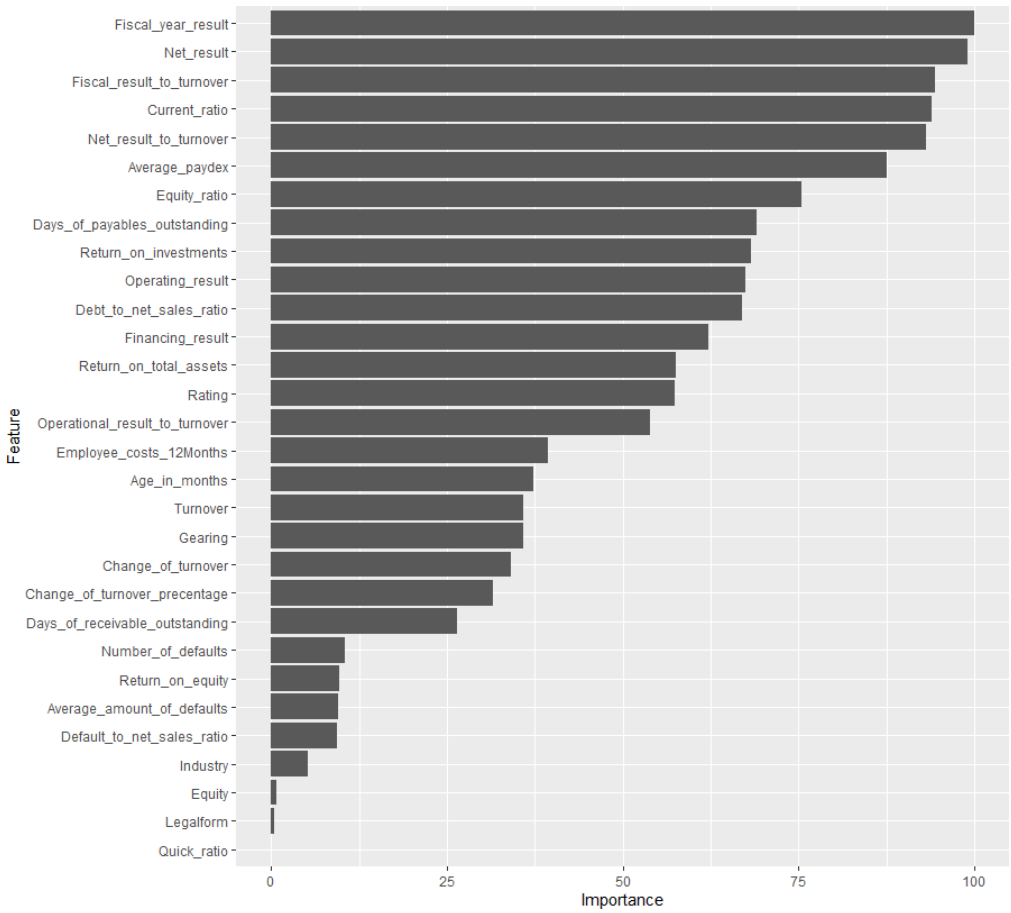

Figure 23. The variable importance in Stepwise LogR model.

The resulting ROC AUC, PR AUC and brier score are presented below in Table 15. With the hold out sample we can confirm that the model performance was good despite the vast amount of predictor variables.

Table 15. The performance metrics of Stepwise LogR model.

Metric	One year before	Two years before
ROC AUC	0,92	0,89
PR AUC	0,86	0,70
Brier score	0,09	0,10

The ROC and PR curves are presented below in Figure 25 and Figure **24**. The shapes of these curves can be considered good.

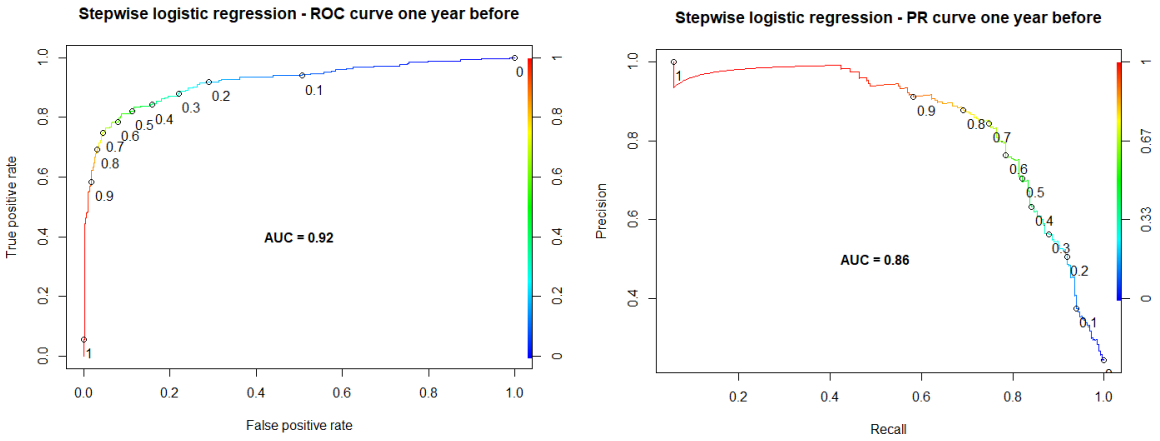

Figure 25. Stepwise LogR model's ROC and PR curves – one year before dataset.

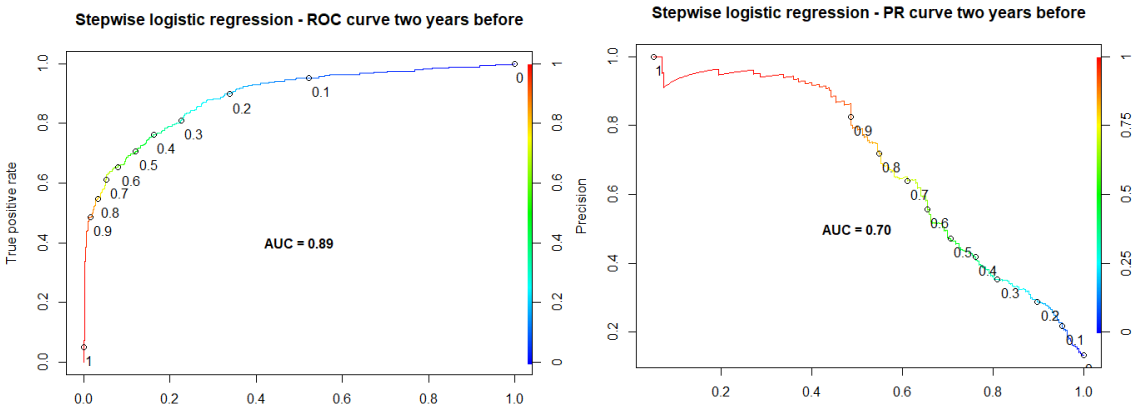

Figure 24. Stepwise LogR model's ROC and PR curves – two yeats before dataset.

5.2.2 Penalized Logistic Regression with LASSO

Least absolute shrinkage and selection operator (LASSO) was used for predictor variable selection. The model was built with glmnet r-package. The hyperparameter optimization was done with testing Lambda (λ) values ranging from $\lambda=10^{10}$ to $\lambda=10^{-2}$ to cover all possibilities between the null model containing only the intercept and the least squares fit. ROC was used to select the optimal model and λ value. The highest ROC was achieved with lambda of 0.01747528.

The predictor variables according to importance are presented in Figure 26. The final model included 21 predictors into the model and is therefore much less complex than the model selected by the stepwise selection. Lower number of predictors resulted in minor performance decrease with the hold-out sample in comparison to stepwise method. Notably, the payment behaviour had the highest relative importance in the model, but external factors number of defaults and default to turnover ratio were left out of the model. In addition, external ratings A-AA were left out of the model. The resulting ROC AUC, PR AUC and brier score are presented below in . The ROC and PR curves are presented in Figure 28 and Figure 27 and the shapes of the curves can be considered good.

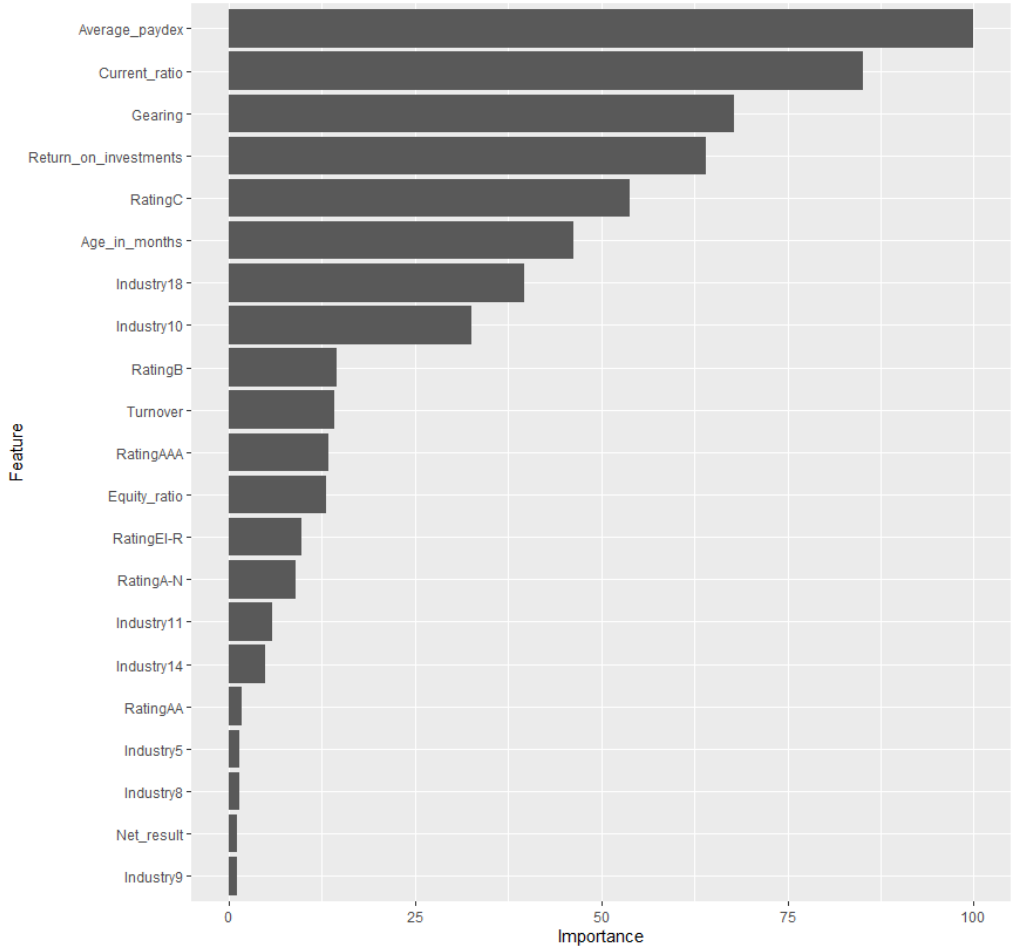

Figure 26. The predictor importance in LASSO LogR model.

Table 16. The performance metrics of the LASSO LogR model.

Metric	One year before	Two years before
ROC AUC	0,91	0,86
PR AUC	0,81	0,58
Brier score	0,11	0,13

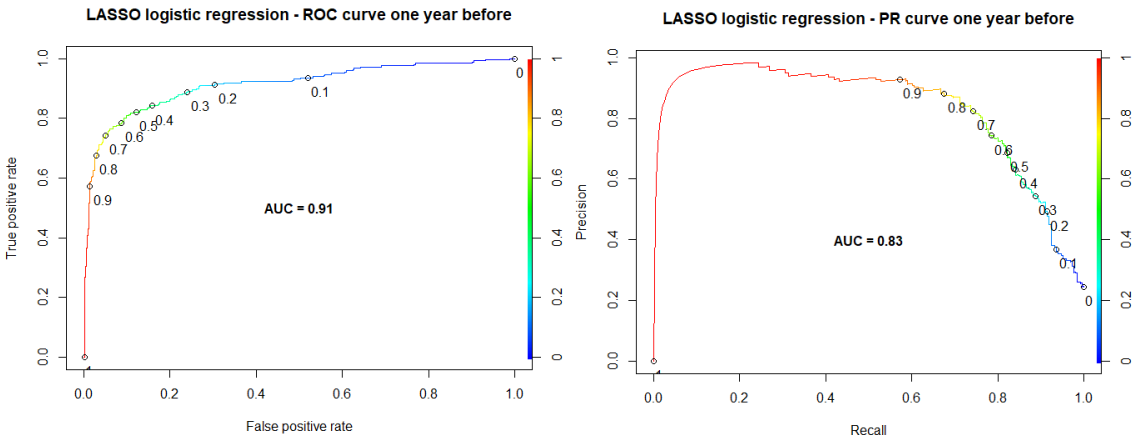

Figure 28. LASSO LogR model's ROC and PR curves – one year before dataset.

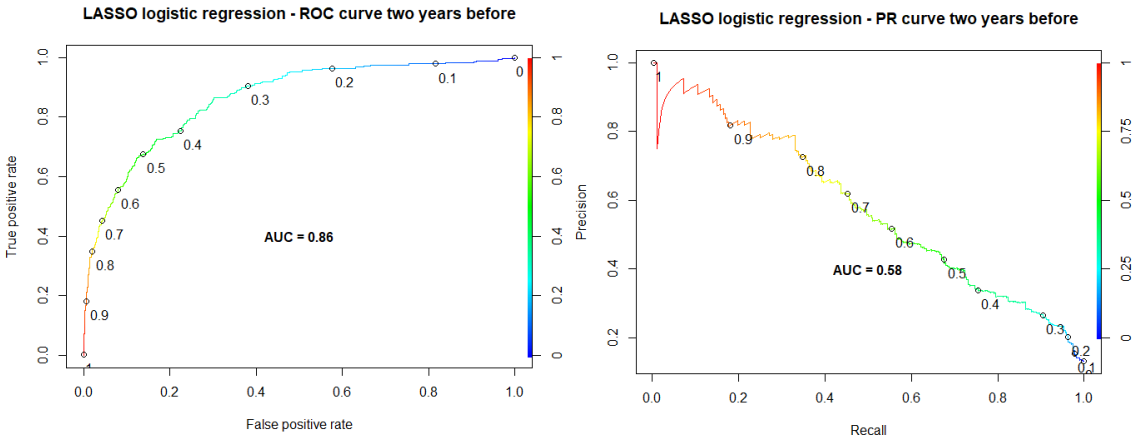

Figure 27. LASSO LR model's ROC and PR curves – two years before dataset.

5.3 Decision Tree Analysis CART

Classification & Recession tree (CART) was built using r-package rpart. ROC value was used to select the optimal pruning hyperparameter called cp. The complexity parameter (cp) in rpart controls the minimum improvement needed at each node of the model. The best performance was reached with cp of 0,01.

The resulting tree presented in Figure 29 can be considered simple and comprehensive. Decision trees can help to interpret the underlying relationships between the variables. From the tree we can learn that payment behavior and gearing have a relationship: longer payment delays (low paydex) result in higher default risk even with lower gearing. This of course logical, as indebted companies who have difficulties to pay invoices clearly have solvency challenges. A rather counterintuitive finding is that very high return on total assets is a sign of default risk. In fact, the dataset has 63 defaulted companies with return on assets of more than 424%. When analyzing these individual companies' financial statements, it was found that abnormally high return on total assets is a result of selling company's assets due to insolvency challenges. This improves company's result and decreases the amount of assets resulting in high return on assets, however without the assets, business cannot continue.

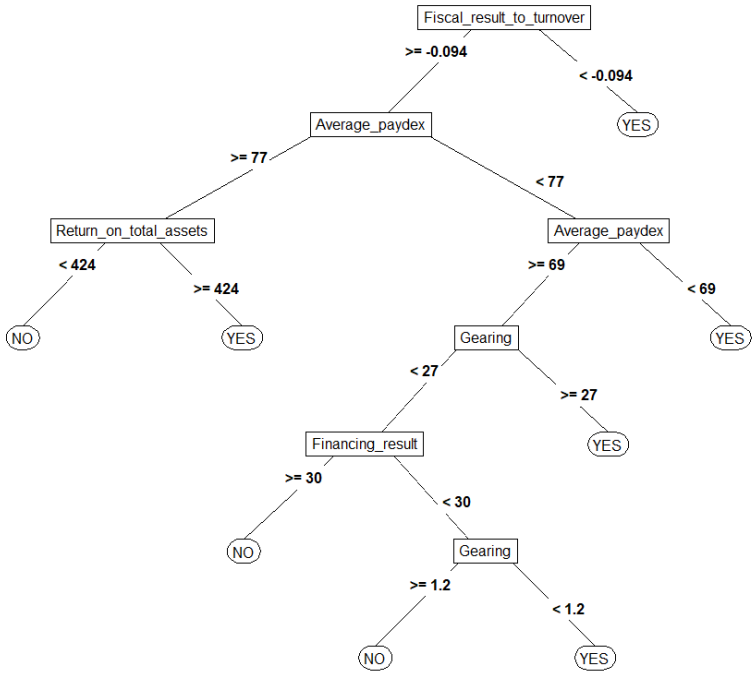

Figure 29. The decision tree of the CART model. Yes = default, No = no default.

Variable importance for decision tree is presented in Figure 30. It should be noted, the final model did not include all variables to the best model presented in Figure 29.

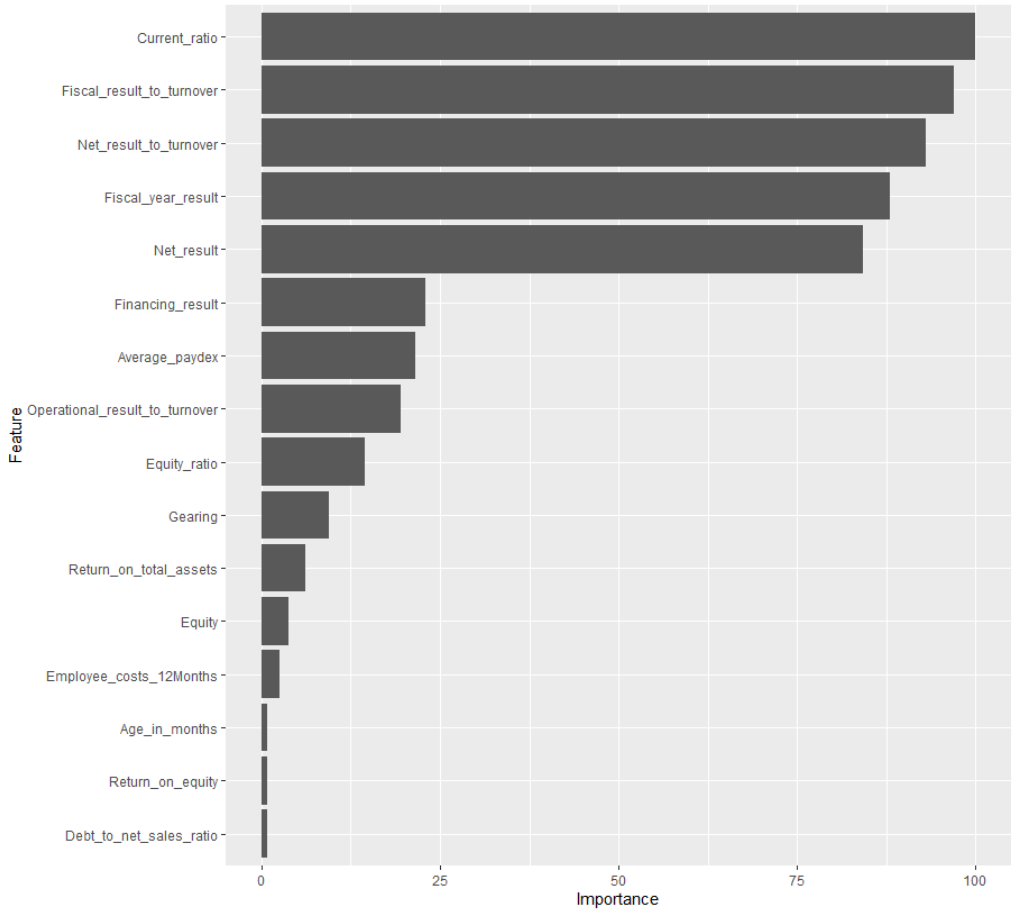

Figure 30. The predictor importance in CART model.

The performance metrics of the CART model are presented below in Table 17. The resulting model performs better than the baseline set by the external rating, however while the ROC AUC is close to lasso logistic regression and MDA models' performance, the Precision-Recall AUC is significantly lower. The Brier score is similar to LASSO logistic regression, but significantly higher than with the MDA model. The ROC and PR curves are presented in Figure 32 and Figure 31 for each dataset. The shape of the curves can be considered good.

Table 17. The performance metrics of the CART model.

Metric	One year before	Two years before
ROC AUC	0,90	0,85
PR AUC	0,75	0,51
Brier score	0,11	0,12

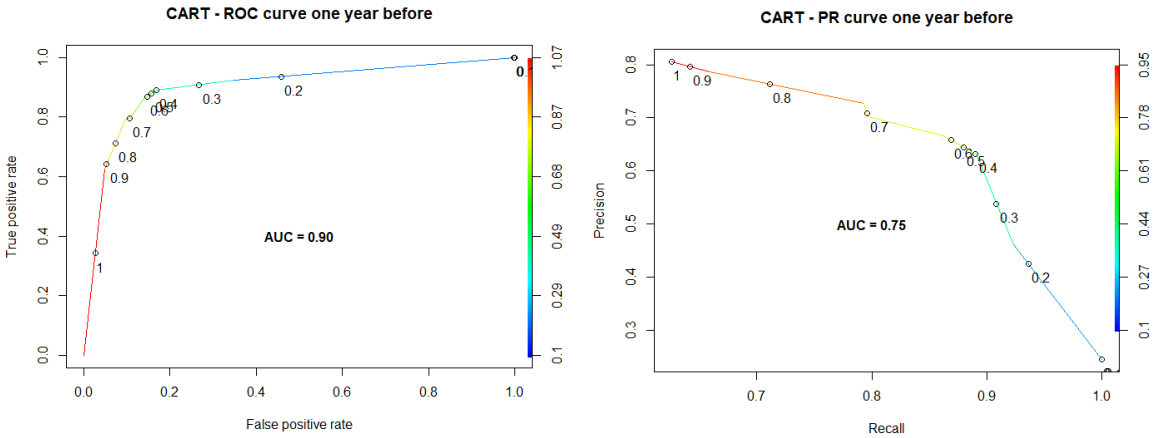

Figure 32. CART model's ROC and PR curves – one year before dataset.

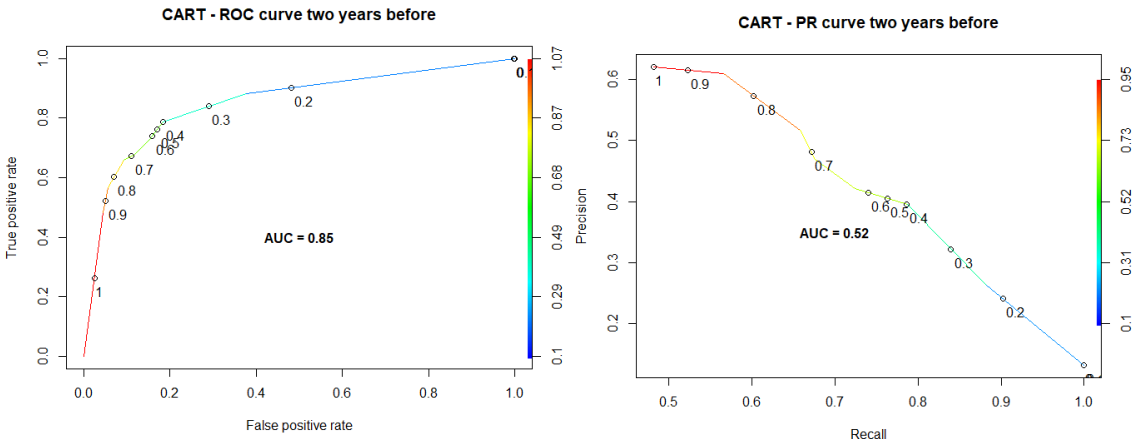

Figure 31. CART model's ROC and PR curves – two years before dataset.

5.4 Random Forest

Random forest model was built using MASS r-package's rf method. The hyper parameter optimization of mtrees was tested with values between 1 to 15, and ROC AUC was used to select the optimal hyperparameter. The final value used for the model was mtry = 6.

Variable importance in Random Forest model is presented in **Figure 33**.

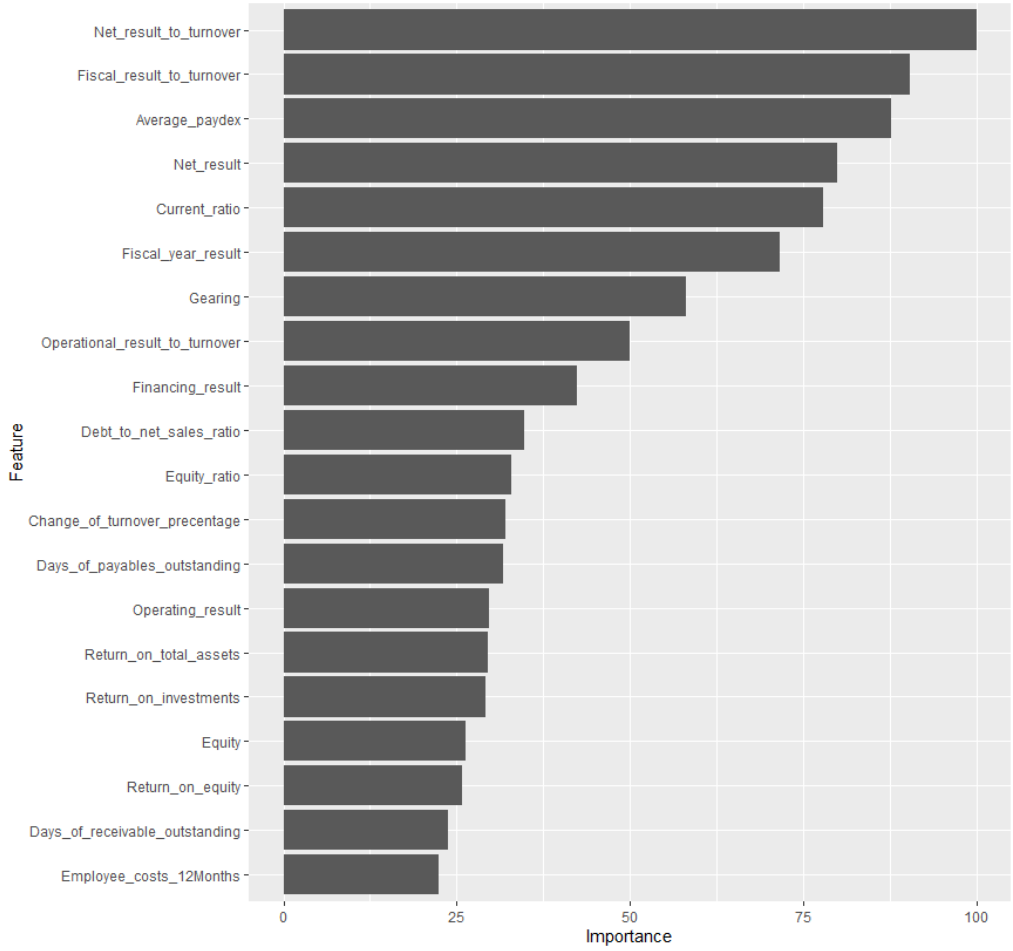

Figure 33. The predictor importance in Random Forest model.

The resulting ROC AUC, PR AUC and Brier score for both datasets are presented below in Table 18. The model performs extremely well in all the selected metrics, significantly better than the baseline, MDA, LASSO logistic regression and CART decision tree. The ROC and PR curves are presented in Figure 34 and Figure 35. The shape of the curves can be considered excellent.

Table 18. The performance metrics of the Random Forest model.

Metric	One year before	Two years before
ROC AUC	0,97	0,93
PR AUC	0,92	0,93
Brier score	0,06	0,06

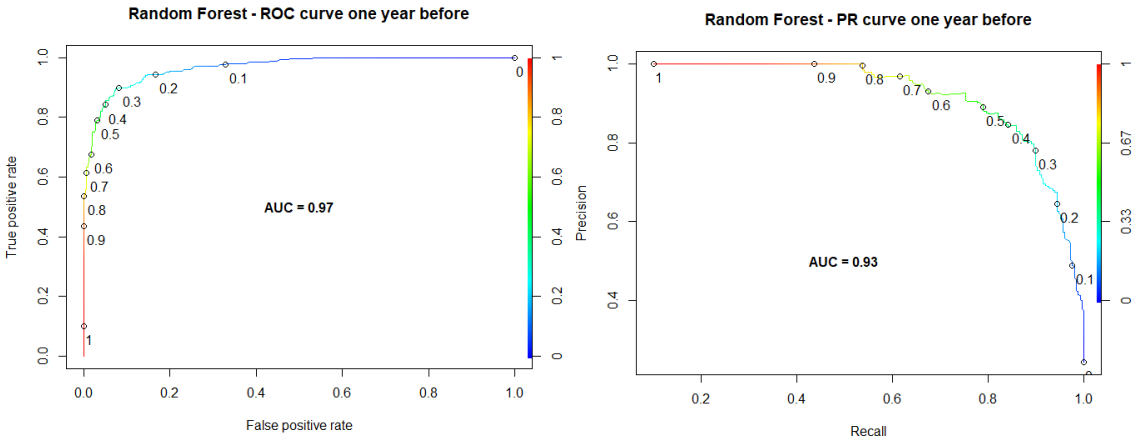

Figure 34. Random Forest model's ROC and PR curves – one year before dataset.

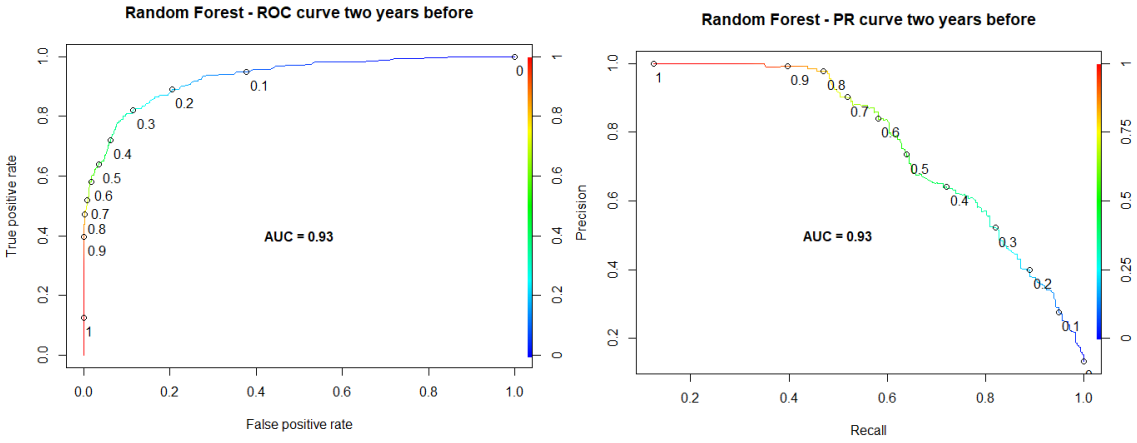

Figure 35. Random Forest model's ROC and PR curves – two years before dataset.

5.5 AdaBoost

Boosted ensembled decision tree, was built with r-package fastAdaboost with method adaboost.m1. The hyperparameter optimization was done by adjusting nIter, the number of weak classifiers. The model was optimized with maximizing ROC value and nIter parameter 150 gave the best performing model.

The resulting model is fairly complex. The relative predictor variable importance in the AdaBoost model presented in Figure 36.

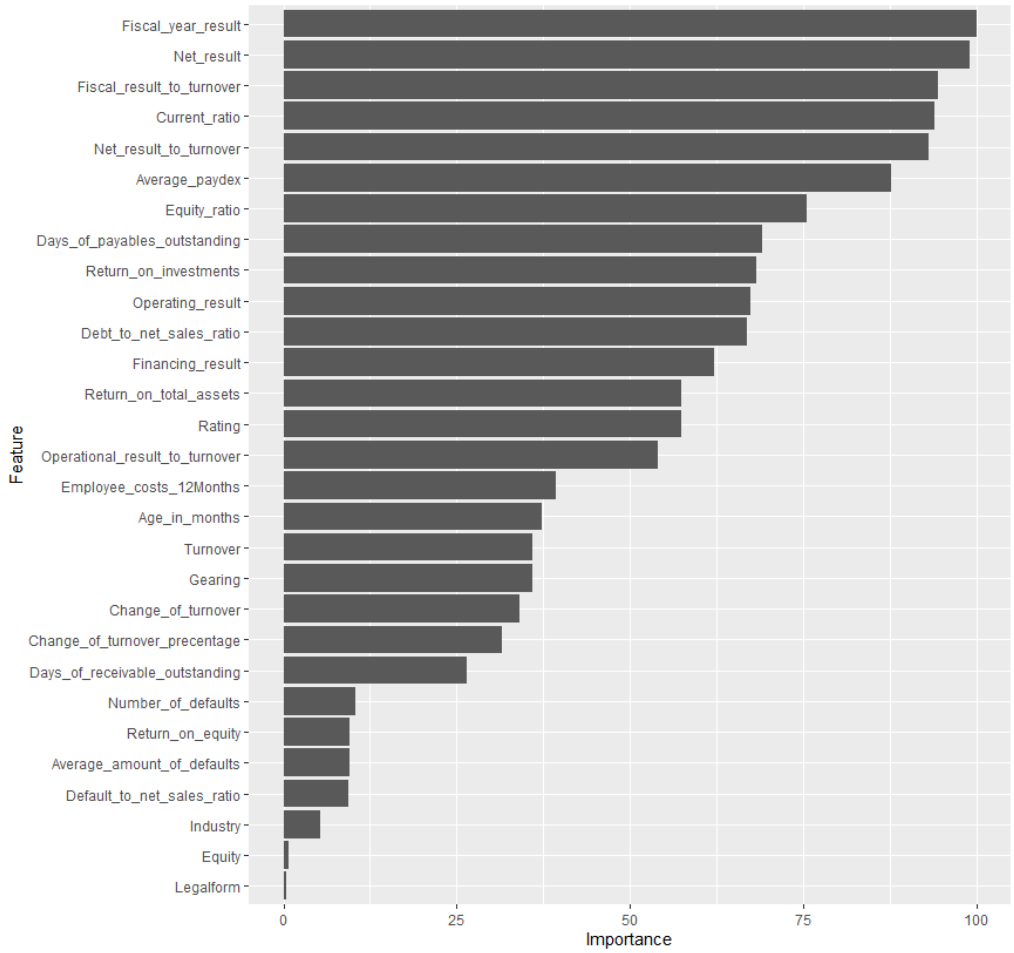

Figure 36. The predictor importance in the AdaBoost model.

The performance metrics of the AdaBoost for both datasets are presented below in . The model performs better than the baseline and all previously tried models in all the selected metrics. The ROC and PR curves are presented in Figure 38 and Figure 37 for each dataset. The shape of the curves can be considered excellent.

Table 19. The performance metrics of the AdaBoost model.

Metric	One year before	Two years before
ROC AUC	0,98	0,95
PR AUC	0,95	0,86
Brier score	0,05	0,05

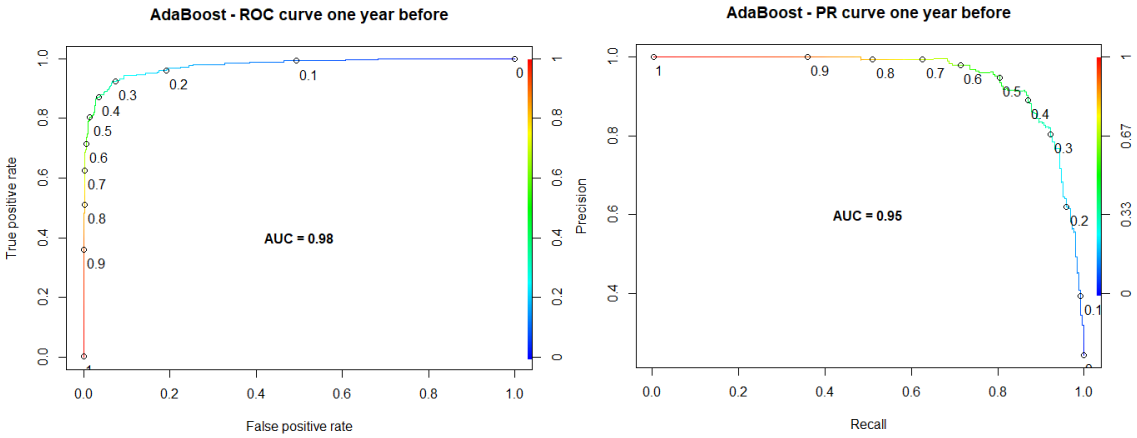

Figure 38. AdaBoost model's ROC and PR curves – one year before dataset.

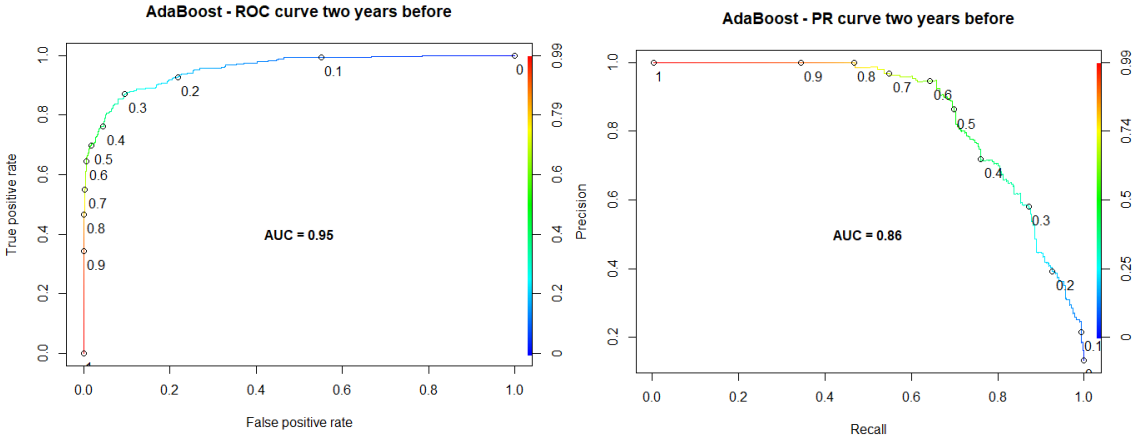

Figure 37. AdaBoost model's ROC and PR curves – two years before dataset.

5.6 Support Vector Machine

Support Vector Machine (SVM) was built using r-package kernlab. The hyperparameter optimization for the SVM model is done by selecting the kernel and hyperparameters. There are many kernels available, but in this study the most commonly used kernels linear, radial and polynomial were tested.

For linear SVM, the optimization is done by finding the optimal value for the cost regularization parameter C, which applies penalty for misclassification. ROC was used to

select the optimal hyperparameter. The highest ROC was reached with the C value of 1.263158. The performance metrics are presented in Table 20.

Table 20. The performance metrics of the linear SVM kernel.

Metric	One year before	Two years before
ROC AUC	0,913	0,877
PR AUC	0,826	0,625
Brier score	0,100	0,110

For radial kernel, the optimization is done by finding the optimal value for both the cost regularization parameter C and Gamma (called Sigma in kernlab package). A low Gamma may result in a constrained model which cannot capture the complexity of the data while high Gamma may result in overfitting. The performance metrics are presented in Table 21.

Table 21. The performance metrics of the radial SVM kernel.

Metric	One year before	Two years before
ROC AUC	0,907	0,859
PR AUC	0,797	0,595
Brier score	0,101	0,104

The polynomial kernel has optimized for maximum ROC by testing a range of values for degree, which controls the flexibility of the decision boundary and the cost parameter C. The highest ROC was achieved with degree of 1 and C of 0,5. The performance metrics are presented in Table 22.

Table 22. The performance metrics of the polynomial SVM kernel.

Metric	One year before	Two years before
ROC AUC	0,915	0,877
PR AUC	0,827	0,625
Brier score	0,098	0,107

The performance differences between the kernels were minimal. However, polynomial kernel performed slightly better than the second-best model with linear kernel. Therefore, SVM with the polynomial kernel-based model is selected as the final model.

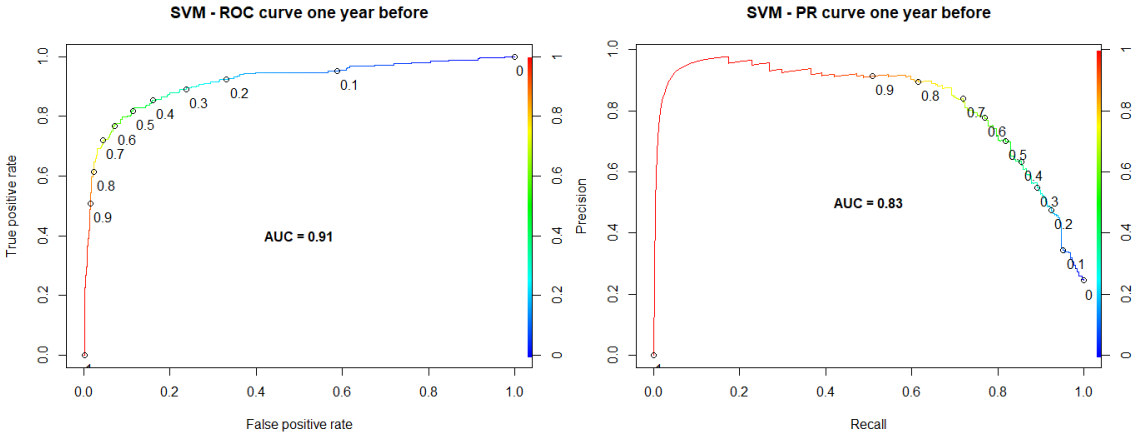

Figure 40. SVM model's ROC and PR curves – one year before dataset.

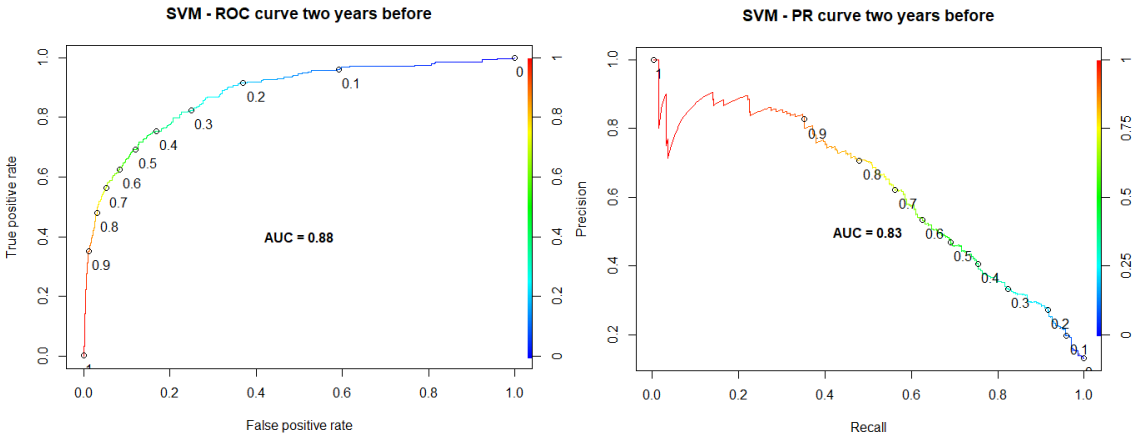

Figure 39. SVM model's ROC and PR curves – two years before dataset.

The ROC and AUC curves for the final model are presented in Figure 40 and Figure 39. The shape of the curves can be considered good, however it should be noted that with the one year before dataset, the precision decreases rapidly with higher threshold levels. The relative predictor significance for SVM model is identical to penalized MDA model and is presented in Figure 41.

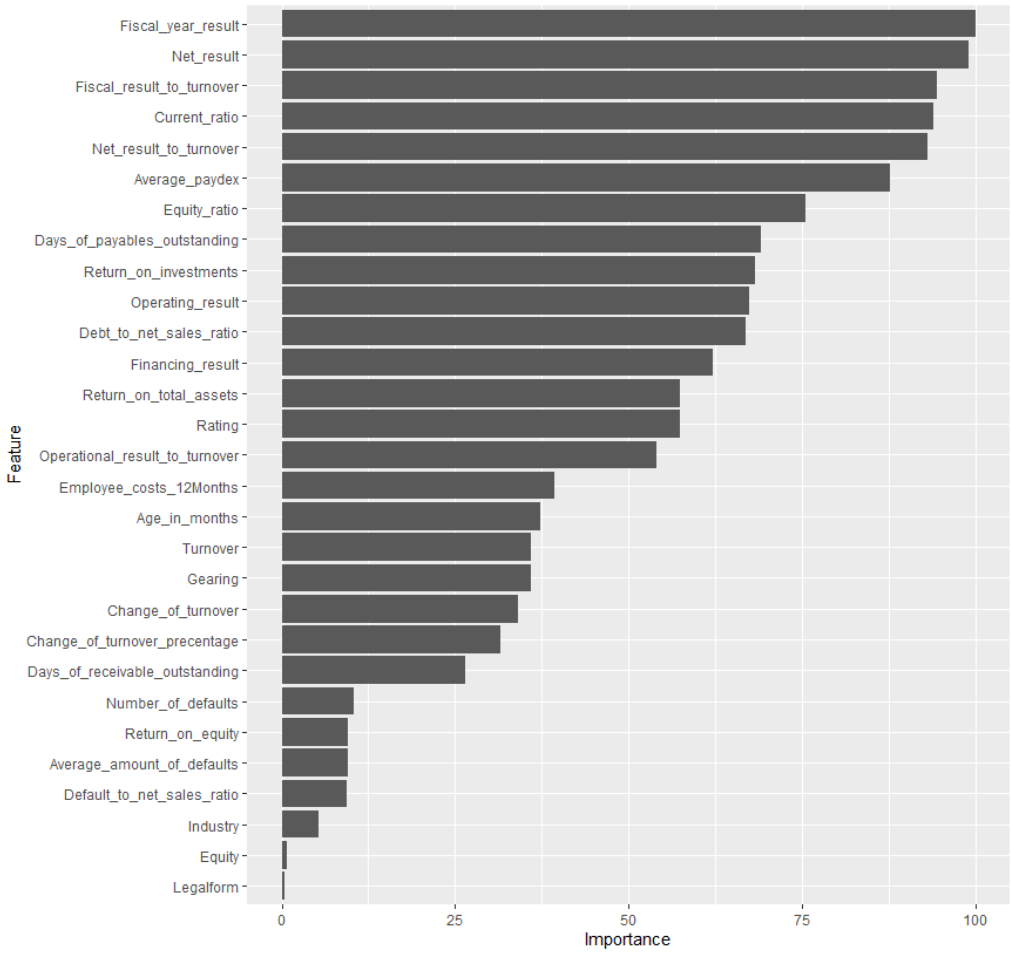

Figure 41. The predictor importance in the final SVM model.

5.7 Neural Network

Neural Network was built using r-package nnet with nnet method. The hyperparameter size was tested with values between 1 to 10 and decay between 0,1 to 0,5. Disabling the over-sampling improved model performance significantly and the ROC-AUC with the hold out test sample improved from 85,6 to 92,7. Therefore the selected model was trained without over-sampling. ROC was used to select the optimal model using the largest value. The optimal model performance was reached with hyperparameter size set to 8 and decay set to 0.5.

Variable importance in the Neural Network model presented in Figure 42. The model found statistical significance in 53 predictors and gearing had the highest relative significance.

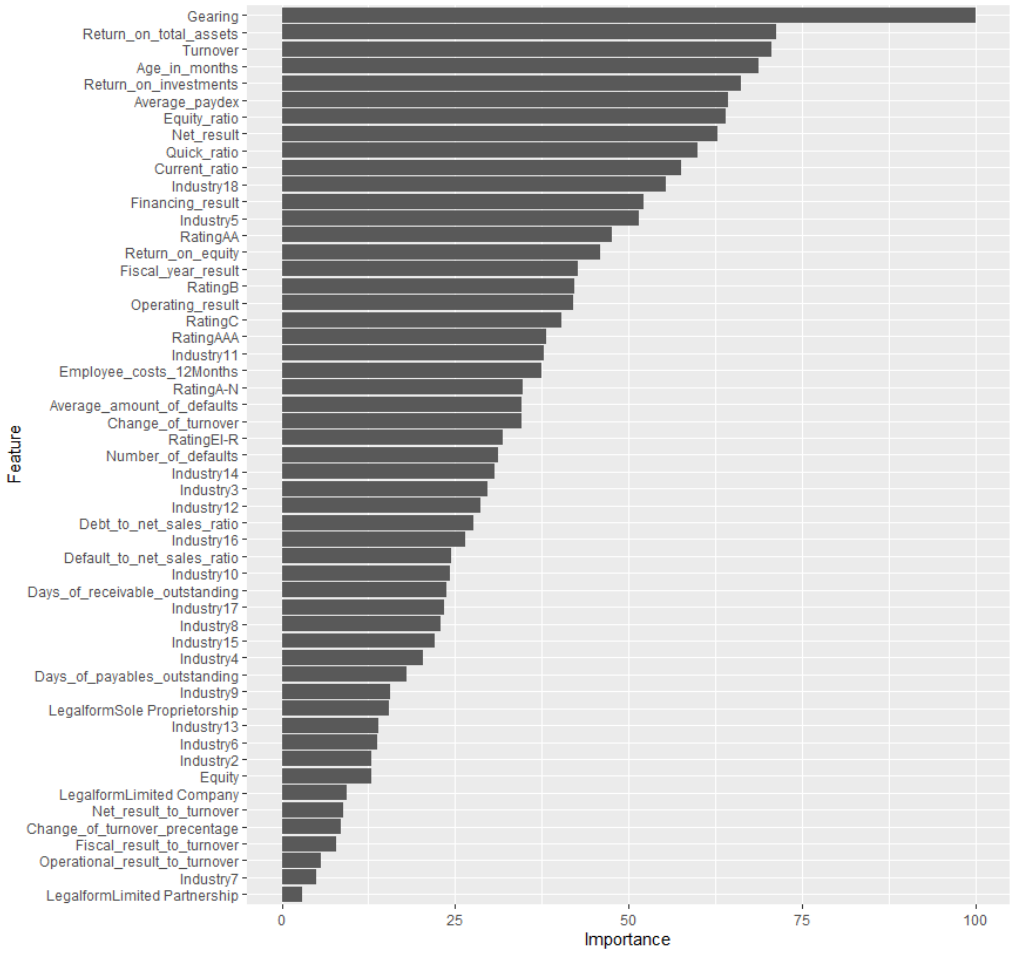

Figure 42. The relative variable importance in the Random Forest model.

The performance metrics for both datasets are presented below in Table 23. The model's performance can be considered good, well above the baseline set by the external rating, but not as well as the highest performing model Adaboost and the second-best model, Random Forest. The ROC and PR curves are presented in Figure 43 and Figure 44 and the shape of the curves can be considered good.

Table 23. The performance metrics of the Neural Network.

Metric	One year before	Two years before
ROC AUC	0,92	0,89
PR AUC	0,87	0,68
Brier score	0,08	0,07

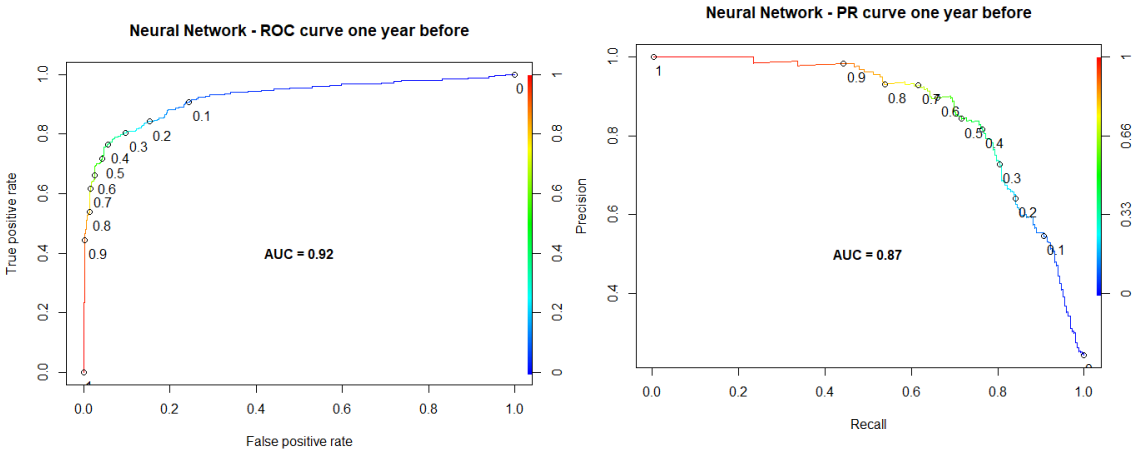

Figure 43. Neural Network model's ROC and PR curves – one year before dataset.

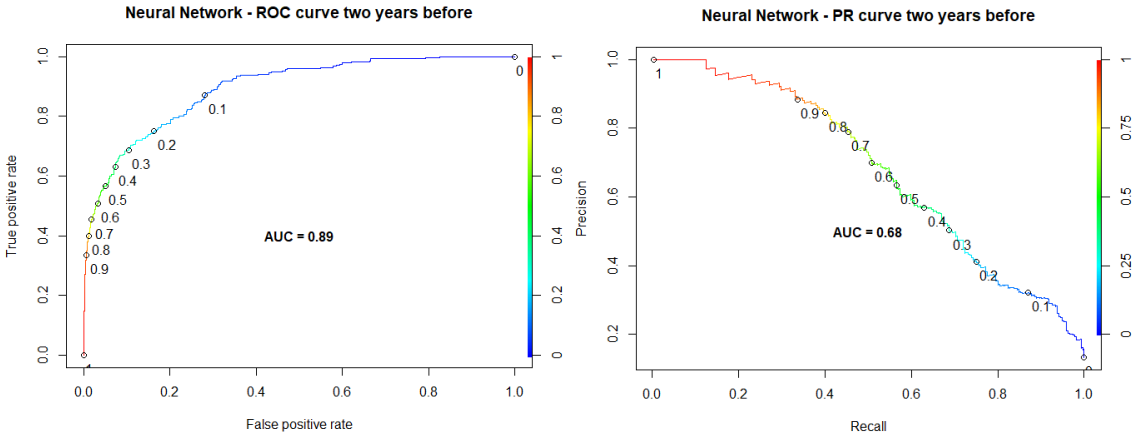

Figure 44. Neural Network model's ROC and PR curves – two years before dataset.

6 Model selection

In summary, all the models performed better than the baseline set by the external rating with the selected metrics. The summary of the performance metrics with the one year before default dataset is shown in Table 24 and with two years before dataset in Table 25.

With the one year before dataset, the Adaboost model performed the best and the Random Forest followed close behind. Both models performed significantly better than all other models in all the selected metrics. The ROC AUC did not have significant differences between the other models. The PR AUC performance was the best with Adaboost and Random Forest followed by the Neural Network and Stepwise LogR. Other models did not perform very well in the PR AUC metric. The Brier scores showed that the Adaboost, Random Forest and Neural Network were confident in the predictions and performed the best, while the LogR models followed close behind.

Table 24. The performance metrics of all the models – one year before dataset.

Metric	External rating	MDA	Stepwise LogR	LASSO LogR	CART	Random Forest	AdaBoost	SVM	Neural Network
ROC AUC	0,79	0,91	0,92	0,91	0,90	0,97	0,98	0,92	0,92
PR AUC	0,55	0,79	0,86	0,81	0,75	0,92	0,95	0,83	0,87
Brier Score	0,17	0,18	0,09	0,11	0,11	0,06	0,05	0,10	0,08

The performance of the models with the two years dataset also showed that all of the models performed better than the baseline set by the external rating. Adaboost was the best performer in ROC AUC and Brier score but performed slightly worse than the Random forest. It should be noted that the PR AUC metric decreased more with this dataset than the ROC AUC metric and Brier Score. It should also be noted that all the models had predictive power also over the longer time horizon.

Table 25. The performance metrics of all the models – two years before dataset.

Metric	External rating	MDA	Stepwise LogR	LASSO LogR	CART	Random Forest	AdaBoost	SVM	Neural Network
ROC AUC	0,74	0,85	0,89	0,86	0,85	0,93	0,95	0,88	0,89
PR AUC	0,53	0,52	0,70	0,58	0,51	0,93	0,86	0,63	0,68
Brier Score	0,11	0,19	0,10	0,13	0,12	0,06	0,05	0,11	0,07

In the model selection, the case company needs to decide between the transparency and the performance of the model. If transparency can be sacrificed, the AdaBoost performed the better than other black box models. However, if transparency is required, the LogR models did perform well. However, also with the LogR models one must balance between the number of predictors and the performance of the model. With reduced number of predictors in LASSO LogR resulted in a minor decrease in the performance of the model in comparison to more complex model created with Stepwise LogR. When comparing the performance of the simplest model CART with transparent MDA, model, the CART would be a clear winner, and the decision tree could even be used in making manual credit decisions due to the simplicity of the model. The black box models may be made more transparent with a surrogate model which is a generally simpler model that are used to explain a more complex model. This is however out of the scope of this thesis.

The recommended transparent model is therefore the LASSO LogR model, but with the significantly higher performance of the Adaboost model with this sample, the case company should consider if full transparency is needed. There are however methods add transparency to the black box models by creating a simpler surrogate model to explain the predictions of the black box model. Creation of the surrogate model is however out of the scope of this study, but it is recommended, that the case company would do further research in this area as the performance of the AdaBoost model is significantly better with the dataset used in this study.

7 Fitting the model into ERP Architecture and Credit Risk Management Process

In this chapter the research question "*How to implement the model effectively to the credit risk management process?"* is answered and the framework for implementing the model into current system and credit management processes is introduced. The chapter consists of four parts, description of the current system architecture, integration to the system architecture, integration to credit risk management processes and dashboard for the credit control team.

7.1 Description of the system architecture

The case company utilizes two separate ERP systems, one for financial accounting including AR, risk management and dunning processes and a separate system for CRM, logistics, inventories, order handling and invoicing. In addition, the case company utilizes multiple data warehouses, analytic platforms, and integrations in the processes. In this chapter the optimal cost benefit balance is sought in the proposed implementation.

7.2 Integration to the system architecture

The data model utilises data from external credit rating agency, including credit rating, payment defaults and financials KPI's based on financial statements. To utilize data for data modelling and analytics purposes, the data is uploaded to Google BigQuery utilizing Google Cloud dataflow and APIs supported by the Credit Rating agency. Internal payment behaviour data from AR is transferred with one day frequency by utilizing current ETL process and the imported to Google BigQuery.

Default prediction model itself is moved to Google Big Cloud compute engine by configuring a virtual machine instance (Ubuntu OS) on Google Cloud, installing R and R Studio Server on the Virtual Machine. The model can be scheduled to run once per day by using cronR r-package. Although, the financial KPI's are updated generally one per year, this way the model can utilize payment behaviour changes as well as external rating changes

to update the PD daily. This way the latest probability for default can be utilized in day to day operations in credit risk management team.

7.3 Integration to Credit Risk Management Processes

Refined credit risk modelling does not add value without leveraging the advanced accuracy and sensitivity in the day to day processes. The increased accuracy of the model enables the organisation to automate the processes further and decreases manual workload. In this chapter the possibilities to implement model into the processes are reviewed.

To utilize the PD's in current risk management processes, the PD's must be uploaded to the ERP system and the processes must be modified to utilize the information in processes to reduce manual handling. The processes can be divided to two sub processes, proactive credit risk mitigation process and reactive collection process.

7.3.1 Credit Risk mitigation process

The purpose of the credit risk handling process is to determine and limit the risk-taking limit for every single customer. Every customer has a set limit for open orders set by the Credit Control team. The limit is set to determine how much risk the case company is willing to take considering the PD for the customer. Credit limit setting is done or every new customer and reviewed based on rating and payment behaviour changes for old customers.

7.3.2 New customer

For every new customer, Credit limit setting is currently done based on the credit scoring decision tree, limit recommendation and expert view. This process functions but adds lead time to order handling process. The suggested change is that all applications and first-time orders are inserted to ERP, also the financial statement information is fetched from credit rating agency utilizing current API connection. This information is stored to the database and credit risk model is run to give the PD and maximum limit for the new

customer. If the limit or order is below the maximum limit, the order is automatically approved and posted for delivery.

In a case the statement is unavailable, and the applicant has requested for a credit limit over a set threshold, the financial statement is asked directly from the applicant and manually imported to the database. After this the limit and PD are determined, and process continues as stated above.

7.3.3 Old customer

For every old customer, the current process is all limit applications and new orders are checked against the set limit in the system. Therefore, keeping the limits up to date is extremely important. Currently this is done by receiving financial statements, rating changes and payment behaviour information via excel file daily. Credit experts are then evaluating the data and changing the limit if needed. This process is slow and prone for errors.

The suggested change is that for each PD, a suggested % of turnover is defined as the maximum open limit automatically. An information of the new limit is sent to customers automatically by the system. Customer can be given certain time to contact the credit department and provide sufficient collateral for higher limit or start paying part of the purchases in advance.

7.3.4 Collection process

For the automated collection process, the dunning parameters can be adjusted according to the PD of the customer. In practise this means sending dunning letters and SMS messages to high risk customers earlier than for the low risk customers. The open receivables can also be sent to collection agency sooner for more efficient debt collection. The recommendation for the stakeholders is that the dunning parameters for each PD is kept up to date in the ERP's dept collection module.

7.4 Credit Risk Dashboard

For analysing case company's customer portfolios credit risk, the proposal is to create a credit risk dashboard for the sales and credit control teams. The dashboard could be created with a self-service business intelligence tool such as Microsoft Power bi or QlikSense. While creating the dashboard is out of the scope of this study, the following KPI's are suggested:

Expected loss (EL) of the selected portfolio: PD x (Exposure – collateral)

Risk to profit ratio in years calculated as: EL / net margin of the customer per annum.

PD change to track the changes in the probability of default for individual customers

Receivable age distribution with the EL for each age group of the receivables

The dashboard can be utilized in credit control team's day to day operations to identify the customers and segments that require immediate attention.

8 Conclusions and recommendations for future research

This chapter summarises the conclusions of the study, answers research questions, discusses limitations of the study, gives suggestions for further research and provides a brief summary of the thesis.

8.1 Summary and Conclusions

In this study, eight different models were tested and compared to the baseline performance set by the external credit rating, that the case company currently utilizes in the credit risk management process. It can be confirmed that all models performed better than the external credit rating in the selected metrics for the classification performance.

Therefore, the answer to the first research question "*Can an in-house machine-learning default risk prediction model, which utilizes both, internal and external predictor variables outperform the currently utilized external credit risk rating?"* is yes, the in-house machine-learning default prediction model can bring additional information and overperform the external rating especially with the lowest credit ratings B and C. However, it should be noted that the purpose of the external credit rating is to give guidance whether to credit the company or not, and the low 12-month PD of 39,5% already suggest, that has been calibrated to be rather risk averse. Therefore, we can confirm, that the in-house model can supplement the external rating and help the credit control team to prioritize their workload, as the high-risk customers can be classified more accurately. In the assessment of the new customer's there is added value, only if the case company is aggressive in their credit strategy and is willing to credit the customers with the low credit rating. Based on the performance metrics, we can conclude that the external credit rating does not underperform and is useful for the purpose it has been built to.

The second research question, *"which model performs the best?"* was answered in chapter 6. The AdaBoost decision tree overperformed all the other models but is a black box model and does not therefore fulfil the set requirement of transparency. The best performing white box model is the logistic regression.

The last research question: "*How to implement the model effectively to the credit risk management process?"* was answered in chapter 7, where the recommendation of how to implement the model into current system architecture and process is given. The integration level may vary, but the proposed solution can be considered cost effective, which was also a set target for the tool.

In summary, all the set targets transparency, accuracy and cost effectiveness were achieved with the proposed model and the model can be implemented to the current systems and processes at minor cost.

8.2 Limitations of the Study

The study utilizes the data already available in the case company's database and therefore the amount of tested predictor variables is limited. Additional variables may improve model's performance.

The used dataset is from a limited time period and majority of the data in one year before dataset from year 2016 and two years before dataset from year 2015. Therefore, this study does not show how the model performs at different economic cycles or in economics shocks. In addition, the dataset used, does not represent a full sample of the corporates of all sizes, legal forms, and industries in Finland. The transportation and construction industries represent majority of the dataset and the limited companies represent more than 99,5% of the data. A larger dataset would be required to study other legal forms and industries.

The predictor variable selection was done by utilizing different methods provided by the r-packages. All these methods have their own limitations. Additional research could be done to test other methods and combinations of the variables to see whether these would result in improved models.

MissForest imputation was used due to missing data in the case company's database. However, the missing data is publicly available, so at additional cost, the missing data could be collected and utilized in the data to avoid possible bias from the imputation. However, imputation is critical part of the process in practise, so that the model is able to deal with the future cases of missing data, as the process is automated.

Due to limited amount of data, the class imbalance issue was dealt with over-sampling, which may result in overfitting with high over-sampling rates and decrease in classifier performance with higher over-sampling rates (Chawla et al., 2002).

8.3 Recommendations for future research

The models were built using the available financial variables, ratio, payment behaviour indicators as well as the characteristics of the company with the currently available data to the case company. The models do not take into account the macroeconomic factors. This could be achieved with introducing external variables such as GDP change, HEX index, inflation and Brent price as well as other industry specific indexes to the model. The impact of macroeconomic factors could be researched with a larger sample size from longer time period with focus on model's performance during global shocks such as the impacts of the 2008 Subprime crisis, COVID-19 as well as the impacts of the Russia's invasion of Ukraine.

In addition, other data imputation methods such as mice could be utilized in the data imputation and for example SMOTE sampling proposed by Chawla et al. (2002) could be utilized to deal with the class imbalance, if larger dataset is not available.

Utilization of the default risk model in pricing is another noteworthy area of research, to leverage the credit risk model even more. The predictions could be utilized to set additional risk premium requirements for different prediction classes.

LIST OF REFERENCES

Adian, I., Doumbia, D., Gregory, N., Ragoussis, A., Reddy, A., & Timmis, J. (2020). Small and medium enterprises in the pandemic.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The journal of finance, 23(4), 589-609.

Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). Journal of banking & finance, 18(3), 505-529.

Altman, E. I. (2013). Predicting financial distress of companies: revisiting the Z-score and ZETA® models. In Handbook of research methods and applications in empirical finance. Edward Elgar Publishing.

Altman, E. I., & Sabato, G. (2013). Modeling credit risk for SMEs: evidence from the US market. In Managing and Measuring Risk: Emerging Global Standards and Regulations After the Financial Crisis (pp. 251-279).

Anandarajan, M., Lee, P., & Anandarajan, A. (2004). Bankruptcy Prediction Using Neural Networks. In Business Intelligence Techniques (pp. 117-132). Springer, Berlin, Heidelberg.

Agarwal, V., & Taffler, R. (2008). Comparing the performance of market-based and accounting-based bankruptcy prediction models. Journal of Banking & Finance, 32(8), 1541-1551.

Aziz, M. A., & Dar, H. A. (2006). Predicting corporate bankruptcy: where we stand?. Corporate Governance: The international journal of business in society.

Bank for International Settlements (2009). Enhancements to the Basel II framework. https://www.bis.org/publ/bcbs157.htm

Bellovary, J. L., Giacomino, D. E., & Akers, M. D. (2007). A review of bankruptcy prediction studies: 1930 to present. Journal of Financial education, 1-42.

Bellovary, J. L., Giacomino, D. E., & Akers, M. D. (2007). A review of bankruptcy prediction studies: 1930 to present. Journal of Financial education, 1-42.

Basel Committee on Banking Supervision (2006). Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework. Comprehensive Version, Bank of International Settlements. http://www.bis.org/publ/bcbs128.htm

Basel Committee on Banking Supervision (2017). Basel III: Finalising post-crisis reforms Bank of International Settlements. https://www.bis.org/bcbs/publ/d424.htm

Beaver, W. H. (1966). Financial ratios as predictors of failure. Journal of accounting research, 71-111.

Berne Union (2021). Yearbook 2021. Available: https://www.berneunion.org/Publications

Bonfim, D. (2009). Credit risk drivers: Evaluating the contribution of firm level information and of macroeconomic dynamics. Journal of banking & finance, 33(2), 281-299.

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR), 49(2), 1-50.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1), 1-3.

Callaghan, J., Murphy, A., & Qian, H. (2015). Third International Conference on Credit Analysis and Risk Management. Cambridge Scholars Publishing.

Chalupka, R., & Kopecsni, J. (2008). Modelling bank loan LGD of corporate and SME segments: A case study (No. 27/2008). IES Working Paper.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

Chen, M. Y. (2011). Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches. Computers & Mathematics with Applications, 62(12), 4514-4524.

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

Ciampi, F., Giannozzi, A., Marzi, G., & Altman, E. I. (2021). Rethinking SME default prediction: a systematic literature review and future perspectives. Scientometrics, 126(3), 2141-2188.

Crouhy, M., Galai, D., & Mark, R. (2000). A comparative analysis of current credit risk models. Journal of Banking & Finance, 24(1-2), 59-117.

Cultrera, L., & Brédart, X. (2016). Bankruptcy prediction: the case of Belgian SMEs. Review of Accounting and Finance.

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).

Dziura, J. D., Post, L. A., Zhao, Q., Fu, Z., & Peduzzi, P. (2013). Strategies for dealing with missing data in clinical trials: from design to analysis. The Yale journal of biology and medicine, 86(3), 343.

Eisenbeis, R. A. (1977). Pitfalls in the application of discriminant analysis in business, finance, and economics. The Journal of Finance, 32(3), 875-900.

Ewert, D. C. (1968). Trade-Credit Management: Selection Of Accounts Receivable Using A Statistical Model. Journal of Finance, 23(5), 891-892.

Fan, A., & Palaniswami, M. (2000, July). Selecting bankruptcy predictors using a support vector machine approach. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (Vol. 6, pp. 354-359). IEEE

The Federal Reserve Board (2006). Basel II Capital Accord Notice of Proposed Rulemaking (NPR) Preamble. https://www.federalreserve.gov/GeneralInfo/Basel2/NPR_20060905/NPR/section_1.htm

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. Pattern recognition letters, 30(1), 27-38.

Finnish Financial Supervisory Authority (2005): Standard 4.4 a Management of credit risk Regulations and guidelines. http://www.finanssivalvonta.fi/en/Regulation/Regulations/New/Documents/4.4a.std1.pdf

Fitzpatrick, P. J. (1932). A comparison of the ratios of successful industrial enterprises with those of failed companies.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

Frydman, H., Altman, E. I., & Kao, D. L. (1985). Introducing recursive partitioning for financial classification: the case of financial distress. The journal of finance, 40(1), 269-291.

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. Annual review of psychology, 60, 549-576.

Gevrey, M., Dimopoulos, I., & Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological modelling, 160(3), 249-264.

Gosiewska, A., & Biecek, P. (2020). Lifting Interpretability-Performance Trade-off via Automated Feature Engineering. arXiv preprint arXiv:2002.04267.

Gong, M. (2021). A novel performance measure for machine learning classification. International Journal of Managing Information Technology (IJMIT) Vol, 13.

Grunert, J., & Weber, M. (2009). Recovery rates of commercial lending: Empirical evidence for German companies. Journal of Banking & Finance, 33(3), 505-513.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society: Series A (Statistics in Society), 160(3), 523-541.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2004). A practical guide to support vector classification.

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). Automated machine learning: methods, systems, challenges (p. 219). Springer Nature.

Joshi, A. (2020). Machine learning and artificial intelligence (1st ed. 2020.). Springer International Publishing.

Kotu, V., & Deshpande, B. (2014). Predictive analytics and data mining: concepts and practice with rapidminer. Morgan Kaufmann.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).

Kubat, M., & Kubat. (2017). An introduction to machine learning (Vol. 2). Cham, Switzerland: Springer International Publishing.

Little, R. J., D'Agostino, R., Cohen, M. L., Dickersin, K., Emerson, S. S., Farrar, J. T., ... & Stern, H. (2012). The prevention and treatment of missing data in clinical trials. New England Journal of Medicine, 367(14), 1355-1360.

Li, X., Ergu, D., Zhang, D., Qiu, D., Cai, Y., & Ma, B. (2022). Prediction of loan default based on multi-model fusion. Procedia Computer Science, 199, 757-764.

Lee, J. W., Lee, W. K., & Sohn, S. Y. (2021). Graph convolutional network-based credit default prediction utilizing three types of virtual distances among borrowers. Expert Systems with Applications, 168, 114411.

Leo, M., Sharma, S., & Maddulety, K. (2019). Machine learning in banking risk management: A literature review. Risks, 7(1), 29.

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

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

Moula, F. E., Guotai, C., & Abedin, M. Z. (2017). Credit default prediction modeling: an application of support vector machine. Risk Management, 19(2), 158-187.

Oba, S., Sato, M. A., Takemasa, I., Monden, M., Matsubara, K. I., & Ishii, S. (2003). A Bayesian missing value estimation method for gene expression profile data. Bioinformatics, 19(16), 2088-2096.

Ong, M.K. (2007). The Basel Handbook. Risk books, a Division of Incisive Financial Publishing. KPMG, 2nd edition

Rubin, D. B. (1976). Inference and missing data. Biometrika, 63(3), 581-592.

Peugh, J. L., & Enders, C. K. (2004). Missing data in educational research: A review of reporting practices and suggestions for improvement. Review of educational research, 74(4), 525-556.

Rosenberg, E., & Gleit, A. (1994). Quantitative methods in credit management: a survey. Operations research, 42(4), 589-613.

Schafer, J. L., & Graham, J. W. (2002). Missing data: our view of the state of the art. Psychological methods, 7(2), 147.

Schmit, M. (2004). Credit risk in the leasing industry. Journal of banking & finance, 28(4), 811-833.

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O., & Hemingway, H. (2014). Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. American journal of epidemiology, 179(6), 764-774.

Shin, K. S., Lee, T. S., & Kim, H. J. (2005). An application of support vector machines in bankruptcy prediction model. Expert systems with applications, 28(1), 127-135.

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

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1), 112-118.

Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., ... & Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. Bmj, 338.

Shetty, S., & Vincent, T. N. (2021). Corporate Default Prediction Model: Evidence from the Indian Industrial Sector. Vision.

Ramosaj, B., & Pauly, M. (2019). Predicting missing values: a comparative study on non-parametric approaches for imputation. Computational Statistics, 34(4), 1741-1764.

Tang, F., & Ishwaran, H. (2017). Random forest missing data algorithms. Statistical Analysis and Data Mining: The ASA Data Science Journal, 10(6), 363-377.

Tharwat, A. (2020). Classification assessment methods. Applied Computing and Informatics.

Van Liebergen, B. (2017). Machine learning: a revolution in risk management and compliance?. Journal of Financial Transformation, 45, 60-67.

Vapnik, V. (1999). The nature of statistical learning theory. Springer science & business media.

Waljee, A. K., Mukherjee, A., Singal, A. G., Zhang, Y., Warren, J., Balis, U., ... & Higgins, P. D. (2013). Comparison of imputation methods for missing laboratory data in medicine. BMJ open, 3(8).

Wilkinson, L. (1999). Statistical methods in psychology journals: Guidelines and explanations. American psychologist, 54(8), 594.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. Journal of Financial and Quantitative Analysis, 15(3), 757-770.

Yu, L., Wang, S., & Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. Expert systems with applications, 34(2), 1434-1444.

9 Appendices

9.1 Descriptive statistics imputed datasets

Descriptive statistics not defaulted companies - one year before dataset

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	2,536	77.2	4.6	18.0	92.2
Number_of_defaults	2,536	0.1	1.0	0	46
Average_amount_of_defaults	2,536	363.3	4,768.8	0.0	162,201.0
Turnover	2,536	7,563.7	37,478.9	−44	900,402
Change_of_turnover	2,536	494.3	6,177.2	−67,586	217,911
Employee_costs_12Months	2,536	1,409.2	7,657.3	0.0	201,880.0
Operating_result	2,536	499.5	6,039.1	−16,429	265,721
Net_result	2,536	375.8	5,420.1	−48,535	223,329
Financing_result	2,536	682.9	6,397.5	−38,999	253,478
Fiscal_year_result	2,536	264.0	5,476.2	−48,530	240,588
Return_on_total_assets	2,536	88.2	152.7	−1,558.0	1,243.0
Return_on_investments	2,536	123.9	251.1	−3,195.0	3,591.0
Return_on_equity	2,536	−6.0	570.1	−13,050.0	9,633.3
Equity	2,536	2,514.0	21,226.1	−16,725	621,102
Equity_ratio	2,536	292.3	339.2	−7,517	997
Debt_to_net_sales_ratio	2,536	172.2	5,345.6	−263,777.3	30,000.0
Gearing	2,536	6.2	65.6	−135.7	2,491.0
Quick_ratio	2,536	4.1	2.9	0.0	9.9
Current_ratio	2,536	1.3	1.2	0.0	9.8
Days_of_receivable_outstanding	2,536	109.5	333.8	0.0	4,581.6
Days_of_payables_outstanding	2,536	4,710.4	26,672.4	−576.1	782,925.0
Age_in_months	2,536	209.3	166.9	6	1,400
Default_to_net_sales_ratio	2,536	0.3	5.5	0.0	229.3
Operational_result_to_turnover	2,536	0.1	0.8	−9.0	38.5
Net_result_to_turnover	2,536	0.4	16.7	−9.0	830.0
Fiscal_result_to_turnover	2,536	0.4	16.8	−9.0	838.9
Change_of_turnover_precentage	2,536	−0.1	9.0	−451.0	1.0

Descriptive statistics defaulted companies - one year before dataset

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	823	70.7	8.4	11.5	87.9
Number_of_defaults	823	0.5	2.4	0	38
Average_amount_of_defaults	823	641.8	4,611.7	0.0	84,134.0
Turnover	823	2,326.2	14,584.7	−6	320,829
Change_of_turnover	823	−1,806.1	41,089.7	−992,608	85,025
Employee_costs_12Months	823	1,252.0	1,578.4	−19.0	16,535.2
Operating_result	823	−333.3	1,040.5	−9,486.7	3,823.1
Net_result	823	−810.6	4,033.4	−95,536.0	3,288.0
Financing_result	823	−386.9	3,980.4	−95,536.0	5,464.3
Fiscal_year_result	823	−836.9	4,039.2	−95,536.0	3,275.0
Return_on_total_assets	823	−182.9	736.0	−7,875	1,946
Return_on_investments	823	−422.5	999.6	−10,267.0	3,650.1
Return_on_equity	823	−271.9	1,144.9	−11,400.0	3,400.0
Equity	823	1,085.2	2,454.5	−3,546.0	34,349.6
Equity_ratio	823	−235.6	949.8	−9,105	949
Debt_to_net_sales_ratio	823	355.3	2,771.5	−47,080.8	13,247.8
Gearing	823	294.1	652.4	−2,910.0	5,166.9
Quick_ratio	823	4.0	2.9	0.0	9.9
Current_ratio	823	0.4	0.7	0.0	6.7
Days_of_receivable_outstanding	823	98.2	261.7	−121.0	3,674.2
Days_of_payables_outstanding	823	5,660.4	15,121.7	0.0	166,364.3
Age_in_months	823	144.5	133.5	6	1,144
Default_to_net_sales_ratio	823	1.5	21.5	0.0	605.3
Operational_result_to_turnover	823	−21.7	412.6	−9,486.7	314.1
Net_result_to_turnover	823	−47.4	876.0	−19,837.7	218.3
Fiscal_result_to_turnover	823	−51.0	932.7	−19,405.9	195.9
Change_of_turnover_precentage	823	−1,951.4	40,643.4	−992,608.0	196.0

Descriptive statistics not defaulted companies - two years before dataset

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	1,788	77.0	4.4	16.5	99.0
Number_of_defaults	1,788	0.04	0.3	0	7
Average_amount_of_defaults	1,788	167.0	3,164.1	0.0	118,122.0
Turnover	1,788	8,634.5	41,423.6	0	833,892
Change_of_turnover	1,788	455.5	6,904.5	−106,671	183,202
Employee_costs_12Months	1,788	1,638.9	9,010.6	0.0	223,601.0
Operating_result	1,788	510.6	5,620.4	−7,168	213,014
Net_result	1,788	383.1	4,975.5	−26,459	181,945
Financing_result	1,788	732.8	6,132.7	−19,859	212,513
Fiscal_year_result	1,788	267.9	5,162.9	−28,097	196,897
Return_on_total_assets	1,788	92.1	183.3	−2,375	1,786
Return_on_investments	1,788	126.2	260.8	−3,029	2,759
Return_on_equity	1,788	−12.6	1,113.5	−32,100.0	8,300.0
Equity	1,788	2,752.2	19,772.8	−1,699	631,172
Equity_ratio	1,788	288.5	366.5	−7,828	995
Debt_to_net_sales_ratio	1,788	251.8	1,132.8	−389.6	29,300.0
Gearing	1,788	7.8	98.1	−145.0	3,486.0
Quick_ratio	1,788	4.1	2.9	−0.1	9.9
Current_ratio	1,788	1.3	1.2	0.0	9.1
Days_of_receivable_outstanding	1,788	83.4	263.2	0.0	4,186.8
Days_of_payables_outstanding	1,788	2,438.6	12,854.8	−24,455.0	153,344.9
Age_in_months	1,788	204.1	170.2	0	1,388
Default_to_net_sales_ratio	1,788	0.1	2.1	0.0	67.7
Operational_result_to_turnover	1,788	0.1	0.3	−8.0	5.0
Net_result_to_turnover	1,788	0.03	0.2	−8.0	4.0
Fiscal_result_to_turnover	1,788	0.03	0.2	−8.0	4.0
Change_of_turnover_precentage	1,788	0.03	0.7	−22.6	1.0

Descriptive statistics defaulted companies - two years before dataset

Statistic	N	Mean	St. Dev.	Min	Max
Average_paydex	272	72.2	6.3	17.0	87.9
Number_of_defaults	272	0.01	0.2	0	3
Average_amount_of_defaults	272	6.3	103.4	0.0	1,705.7
Turnover	272	1,875.4	4,549.5	0	35,607
Change_of_turnover	272	183.3	1,378.6	−2,214	17,667
Employee_costs_12Months	272	1,017.3	1,162.2	0.0	7,557.0
Operating_result	272	−14.4	598.1	−4,597.0	3,233.1
Net_result	272	−322.1	663.6	−4,965.0	2,518.4
Financing_result	272	118.8	606.3	−4,193.0	3,877.6
Fiscal_year_result	272	−352.8	703.5	−4,966.0	1,636.8
Return_on_total_assets	272	23.5	389.9	−3,585	1,579
Return_on_investments	272	236.7	4,722.5	−3,333.0	76,985.0
Return_on_equity	272	−8.9	616.2	−3,643.4	1,893.9
Equity	272	999.8	1,775.8	−498.5	14,396.0
Equity_ratio	272	10.3	629.3	−6,600	986
Debt_to_net_sales_ratio	272	496.2	1,303.7	−4,291.5	9,032.7
Gearing	272	286.9	1,015.8	−1,650.0	14,130.0
Quick_ratio	272	4.0	2.9	−0.8	9.9
Current_ratio	272	0.5	0.7	−0.3	4.7
Days_of_receivable_outstanding	272	81.6	196.8	0.0	2,379.4
Days_of_payables_outstanding	272	5,267.0	10,209.7	0.0	97,901.0
Age_in_months	272	140.9	144.4	3	1,132
Default_to_net_sales_ratio	272	0.01	0.1	0.0	2.1
Operational_result_to_turnover	272	−4.0	56.3	−847.3	107.7
Net_result_to_turnover	272	−7.3	67.6	−1,029.1	59.9
Fiscal_result_to_turnover	272	−6.5	57.4	−793.3	76.0
Change_of_turnover_precentage	272	1.5	19.3	−5.7	315.5

9.2 R -code

9.2.1 Data import

```
library(dplyr)
library(tidyverse)
library(lubridate) # dmy function
#combine all defaults data to one table

defaults_1 <- read.csv("previous default data here.csv", sep=",", na.strings=c("","NA"))
defaults_2 <- read.csv("New default data here .csv", sep=",", na.strings=c("","NA"))

all_defaults_paydex <- rbind(defaults_1, defaults_2)


#Format
all_defaults_paydex$Date <-as.Date(all_defaults_paydex$Date)
all_defaults_paydex$Loaddate <- as.Date(all_defaults_paydex$Loaddate)
all_defaults_paydex$REF <- as.factor(all_defaults_paydex$REF)
all_defaults_paydex$Regnum <- as.character(all_defaults_paydex$Regnum)
all_defaults_paydex$PaymDisDate <- as.Date(as.character(all_defaults_paydex$PaymDis-
Date), format = "%Y%m%d")

#Check
str(all_defaults_paydex)
#str(all_defaults_paydex_bind)

default_new1 <- all_defaults_paydex %>%
  mutate(Valuetype = factor(Valuetype)) %>%
  mutate(Descx = factor(Descx)) %>%
  filter(all_defaults_paydex$Descx == "YVK"| all_defaults_paydex$Descx == "SVK"| all_de-
faults_paydex$Descx == "UMP"| all_defaults_paydex$Descx ==
"UM"|all_defaults_paydex$Descx == "UMV"| all_defaults_paydex$Descx == "UMS"| all_de-
faults_paydex$Descx == "ATR"| all_defaults_paydex$Descx == "AST"| all_defaults_pay-
dex$Descx == "LKP"| all_defaults_paydex$Descx == "OSP"| all_defaults_paydex$Descx ==
"TTT"| all_defaults_paydex$Descx == "TK"| all_defaults_paydex$Descx == "MOP"| all_de-
faults_paydex$Descx == "SEL") # Approved defaults
  #filter(all_defaults_paydex$Descx == "ATR" | all_defaults_paydex$Descx == "AST")

#checks
str(default_new1)
nrow(default_new1)
nrow(all_defaults_paydex)


#format and select needed columns
All_defaults_new <- default_new1 %>%
  mutate(business_id = Regnum) %>%
  mutate(PAYMDISDATE = PaymDisDate) %>%
  mutate(PAYMDISAMOUNT = PaymDisAmount) %>%
  select(PAYMDISAMOUNT, PAYMDISDATE, business_id)

nrow(All_defaults_new)

#Remove duplicate rows
library(dplyr)
All_defaults_new <- distinct(All_defaults_new)

nrow(All_defaults_new)
str(All_defaults_new)

write.csv(All_defaults_new, file = "All_defaults_new.csv",row.names=FALSE)


#Filter, convert to correct form and delect
All_paydex_new1 <- all_defaults_paydex %>%
  #filter(all_defaults_paydex$Paydex != "" & all_defaults_paydex$Paydex != "0") %>%
  filter(all_defaults_paydex$Paydex != "") %>%
```

```
  mutate(business_id = Regnum) %>%
  mutate(DATE = Date) %>%
  mutate(paydex = Paydex) %>%
  select(business_id, paydex, DATE)

#remove duplicates
All_paydex_new <- distinct(All_paydex_new1)

#checks
nrow(All_paydex_new1)
nrow(All_paydex_new)
str(All_paydex_new)

write.csv(All_paydex_new, file = "All_paydex_new.csv",row.names=FALSE)



#Filter, convert to correct form and delect
All_rating_new1 <- all_defaults_paydex %>%
  mutate(rating = ifelse(is.na(all_defaults_paydex$Newcreditrating), all_defaults_pay-
dex$Oldcreditrating, all_defaults_paydex$Newcreditrating)) %>%
  mutate(rating = fct_recode(rating, "EI-R" = "EI-")) %>%
  mutate(business_id = Regnum) %>%
  mutate(DATE = Date) %>%
  select(business_id, rating, DATE, Oldcreditrating, Newcreditrating)


str(All_rating_new1)
nrow(All_rating_new1)

#remove duplicates
All_rating_new <- distinct(All_rating_new1)
str(All_rating_new)
nrow(All_rating_new)

write.csv(All_rating_new, file = "All_rating_new.csv",row.names=FALSE)

#load company and financial KPIs
full_dataset <- read.csv("Company information and financial kpis here.csv", sep=";",
colClasses=c("NACE.code.2"="factor")) %>%
  mutate(Fiscal_year = dmy(Fiscal_year)) %>%
  mutate(business_id = as.character(business_id)) %>%
  mutate(NACE.code = as.factor(NACE.code)) %>%
  mutate(status= as.factor(status)) %>%
  mutate(legalform= as.factor(legalform)) %>%
  mutate(turnover= as.numeric(turnover)) %>%
  mutate(change_of_turnover= as.numeric(change_of_turnover)) %>%
  mutate(return_on_equity_P= as.numeric(return_on_equity_P)) %>%
  mutate(equity= as.numeric(equity))  %>%
  mutate(company_name = as.factor(company_name)) %>%
  mutate(NACE.code.format = as.factor(NACE.code.format))

head(full_dataset)

#load defaults data
defaults <- read.csv("All_defaults_new.csv", sep=",")


running_totals<-
  defaults %>%
  group_by(business_id) %>%
  arrange(PAYMDISDATE) %>%
  mutate(
    n_defaults = row_number(), # a running total
    avg_amount_defaults = cummean(PAYMDISAMOUNT) # a cummulative mean
  ) %>%
  ungroup()


full_dataset_defaults <- running_totals%>%
  left_join(full_dataset, by = "business_id") %>% # only common column is ID so you
will get all combos of Date1 & Date2
```

```
  group_by(business_id, Fiscal_year) %>%
  filter(
    PAYMDISDATE < Fiscal_year,
  ) %>%

  filter(
    n_defaults == max(n_defaults) # will be the last row before each Date1
  ) %>%
  ungroup() %>%
  select(business_id, Fiscal_year, n_defaults, avg_amount_defaults) %>% # put the col-
umns in order
  full_join(full_dataset) #%>% # used to bring in the ID/Date combos that didn't have
defaults


paydex <-read.csv("All_paydex_new.csv", sep=",") #uusi datasetti

running_totals_paydex<-
  paydex %>%
  group_by(business_id) %>%
  arrange(DATE) %>%
  mutate(
    n_paydex = row_number(), # a running total
    avg_amount_paydex = cummean(paydex) #cumulative mean
  ) %>%
  ungroup()

full_dataset_defaults_paydex <- running_totals_paydex%>%
  left_join(full_dataset_defaults, by = c("business_id")) %>% # only common column is ID
so you  will get all combos of Date1 & Date2
  group_by(Fiscal_year, business_id) %>%
  filter(
   DATE < Fiscal_year,
  ) %>%
  filter(
    n_paydex == max(n_paydex) # will be the last row before each Date1
  ) %>%
  ungroup()%>%
  select(business_id, Fiscal_year, avg_amount_paydex, n_paydex) %>%
  right_join(full_dataset_defaults, by = c("business_id", "Fiscal_year")) #%>% # used to
bring in the ID/Date combos that didn't have paydex
full_dataset_defaults_paydex[c("n_defaults", "avg_amount_defaults")][is.na(full_da-
taset_defaults_paydex[c("n_defaults", "avg_amount_defaults")])] <- 0

write.csv(full_dataset_defaults_paydex, file = "full_dataset_with_defaults_pay-
dex.csv",row.names=FALSE)


#str(paydex_calc)
str(full_dataset_defaults)
str(full_dataset_defaults_paydex)


#nrow(paydex_calc)
nrow(full_dataset)
nrow(full_dataset_defaults)
nrow(full_dataset_defaults_paydex)


rating <-read.csv("All_rating_new.csv", sep=",")

running_totals_rating<-
  rating %>%
  group_by(business_id) %>%
  arrange(DATE) %>%
  mutate(
    n_rating = row_number(), # a running total
  ) %>%
  ungroup()

full_dataset_defaults_paydex_rating <- running_totals_rating%>%
```

```
  left_join(full_dataset_defaults_paydex, by = c("business_id")) %>% # only common col-
umn is ID so you  will get all combos of Date1 & Date2
  #filter(business_id == "D") %>% # you can run this to check the logic
  group_by(Fiscal_year, business_id) %>%
  filter(
    DATE < Fiscal_year,
  ) %>%
  filter(
    n_rating == max(n_rating) # will be the last row before each Date
  ) %>%
  ungroup()%>%
  select(business_id, Fiscal_year, rating) %>%
  right_join(full_dataset_defaults_paydex, by = c("business_id", "Fiscal_year")) #%>% #
used to bring in the ID/Date combos that didn't have defualts

str(full_dataset_defaults_paydex_rating)


#add NACI description level 2 (industry)
NACE_description <- read.csv("NACE_description.csv", sep=";") %>%
  mutate(Code = as.factor(Code))

NACE_description  <- NACE_description  %>%
  select(Code, Description)  %>%
  rename(NACE.code.2 = Code) %>%
  mutate(NACE.code.2 = as.factor(NACE.code.2))  %>%
  rename(NACE.description = Description)
full_dataset_defaults_paydex_rating  <- full_dataset_defaults_paydex_rating %>%
  left_join(NACE_description, by = c("NACE.code.2"))

#add also NACE level 1 (industry) as too many level 2 categories

NACE_description_level_1 <- read.csv("NACE_description_level1.csv", sep=";", col-
Classes=c("Code"="factor")) %>%
  select(Code,  NACE_description_level_1)  %>%
  mutate(Code = as.factor(Code)) %>%
  rename(NACE.code.2 = Code)


full_dataset_defaults_paydex_rating  <- full_dataset_defaults_paydex_rating %>%
  left_join(NACE_description_level_1 , by = c("NACE.code.2"))

#add ranks to NACE level 1 (industry) categories
full_dataset_defaults_paydex_rating <- full_dataset_defaults_paydex_rating %>%
  mutate(Industry= as.numeric(factor(NACE_description_level_1))) %>%
  mutate(Industry = as.factor(Industry))

str(full_dataset_defaults_paydex_rating)


#convert table

full_dataset_defaults_paydex_rating <- full_dataset_defaults_paydex_rating %>%
  mutate(NACE.description = as.factor(NACE.description)) %>%
  mutate(rating= as.factor(rating))  %>%
  mutate(NACE_description_level_1 = as.factor(NACE_description_level_1)) %>%
  mutate(Default_status_prob = as.numeric(default_status)) %>%
  mutate(default_status = as.factor(default_status))%>%
  mutate(default_status = fct_recode(default_status, "YES" = "1")) %>%
  mutate(default_status = fct_recode(default_status, "NO" = "0"))


str(full_dataset_defaults_paydex_rating)


str(full_dataset_defaults_paydex_rating)
nrow(full_dataset_defaults_paydex_rating)
write.csv(full_dataset_defaults_paydex_rating, file = "full_dataset_with_defaults_pay-
dex_rating.csv",row.names=FALSE)


str(full_dataset_defaults_paydex_rating)
```

```
#Rename dataset
full_dataset_defaults_paydex_rating <- full_dataset_defaults_paydex_rating %>%
  rename(Business_id = business_id) %>%
  rename(Fiscal_year = Fiscal_year) %>%
  rename(Rating = rating) %>%
  rename(Average_paydex = avg_amount_paydex) %>%
  rename(N_paydex = n_paydex) %>%
  rename(Number_of_defaults = n_defaults) %>%
  rename(Average_amount_of_defaults = avg_amount_defaults) %>%
  rename(Default_status = default_status) %>%
  rename(Defaulting_date = defaulting_date) %>%
  rename(Company_name = company_name) %>%
  rename(Status = status) %>%
  rename(Legalform = legalform) %>%
  rename(Date_of_registration = date_of_registration) %>%
  rename(NACE.code = NACE.code) %>%
  rename(NACE.code.format = NACE.code.format) %>%
  rename(NACE.code.2 = NACE.code.2) %>%
  rename(Turnover = turnover) %>%
  rename(Change_of_turnover = change_of_turnover) %>%
  rename(Employee_costs_12Months = employee_costs_12Months) %>%
  rename(Change_of_employee_costs = change_of_employee_costs) %>%
  rename(Change_of_balance_total = change_of_balance_total) %>%
  rename(Operating_result = operating_esult) %>%
  rename(Net_result = net_result) %>%
  rename(Financing_result = financing_result) %>%
  rename(Fiscal_year_result = fiscal_year_result) %>%
  rename(Return_on_total_assets = return_on_total_assets) %>%
  rename(Return_on_investments = return_on_investments) %>%
  rename(Return_on_equity = return_on_equity_P) %>%
  rename(Equity = equity) %>%
  rename(Equity_ratio = equity_ratio) %>%
  rename(Debt_to_net_sales_ratio = debt_to_net_sales_ratio) %>%
  rename(Gearing = gearing_.) %>%
  rename(Quick_ratio = quick_ratio) %>%
  rename(Current_ratio = current_ratio) %>%
  rename(Days_of_receivable_outstanding = days_of_receivable) %>%
  rename(Days_of_payables_outstanding = days_of_payables) %>%
  rename(Number_of_employees = number_of_employees.) %>%
  rename(All_data = All_data.) %>%
  rename(Age_in_months = age_months_in_fiscal_year) %>%
  rename(Registered_months_before_defaulting = registered_months_before_defaulting) %>%
  rename(NACE.description = NACE.description) %>%
  rename(Industry_name = NACE_description_level_1)

#remove useless columns
full_dataset_defaults_paydex_rating$Number_of_employees <- NULL
full_dataset_defaults_paydex_rating$Change_of_employee_costs <- NULL
full_dataset_defaults_paydex_rating$Change_of_balance_total <- NULL
full_dataset_defaults_paydex_rating$N_paydex<- NULL
full_dataset_defaults_paydex_rating$All_data <- NULL
full_dataset_defaults_paydex_rating$NACE.code.format <- NULL
full_dataset_defaults_paydex_rating$NACE.code.2 <- NULL
full_dataset_defaults_paydex_rating$NACE.code.y <- NULL
full_dataset_defaults_paydex_rating$NACE.code <- NULL
full_dataset_defaults_paydex_rating$NACE.description <- NULL
full_dataset_defaults_paydex_rating$N_paydex <- NULL


str(full_dataset_defaults_paydex_rating)

length (unique(full_dataset_defaults_paydex_rating$Business_id))


#str(paydex_calc)
str(full_dataset_defaults)
str(full_dataset_defaults_paydex)
str(full_dataset_defaults_paydex_rating)


#nrow(paydex_calc)
```

```
nrow(full_dataset)
nrow(full_dataset_defaults)
nrow(full_dataset_defaults_paydex)
nrow(full_dataset_defaults_paydex_rating)


#filter only data with 1-100 months before defaulting (in dataset there is data after
defaulting)
full_dataset_defaults_paydex_rating<- full_dataset_defaults_paydex_rating[with(full_da-
taset_defaults_paydex_rating, (Registered_months_before_defaulting >= 1 & Regis-
tered_months_before_defaulting <= 100)), ]

str(full_dataset_defaults_paydex_rating)
nrow(full_dataset_defaults_paydex_rating)
length (unique(full_dataset_defaults_paydex_rating$Business_id))


write.csv(full_dataset_defaults_paydex_rating, file = "full_dataset_with_defaults_pay-
dex_rating.csv",row.names=FALSE)
```

9.2.2 Imputation

```
library(naniar) #for missing data analysis
library(mice) #for imputation
library(Hmisc) # for imputation
library(missForest)# for imputation
#library(kableextra)  #for table
library(doParallel) # for paraler imputation in missforest
library(broom) # for cleaning up data models to merge to tables
library(sjmisc) # for dunction replace column used to combine imputed data and original
dataset
library(xtable) # for latex table

#Count of all NAs per column
na_count_rating <-sapply(full_dataset_defaults_paydex_rating, function(y)
sum(length(which(is.na(y)))))
na_count_rating <- data.frame(na_count_rating)
na_count_rating

full_dataset_defaults_paydex_rating <- as.data.frame(full_dataset_defaults_paydex_rat-
ing)

#split to one and two years before defaulting
full_dataset_defaults_paydex_ratin_one_year_before <- full_dataset_defaults_paydex_rat-
ing[with(full_dataset_defaults_paydex_rating, (Registered_months_before_defaulting  >= 1
& Registered_months_before_defaulting  <= 23)), ]#%>%
#select(default_status, age_months_in_fiscal_year, equity_ratio, quick_ratio, n_de-
faults, avg_amount_paydex, return_on_total_assets)
full_dataset_defaults_paydex_ratin_two_years_before <- full_dataset_defaults_paydex_rat-
ing[with(full_dataset_defaults_paydex_rating, (Registered_months_before_defaulting  >=
24 & Registered_months_before_defaulting <= 36)), ]


#test with Hmisc package if it is feasible to impute data hierarical clustering
plot(naclus(full_dataset_defaults_paydex_rating))
plot(naclus(full_dataset_defaults_paydex_ratin_one_year_before))
plot(naclus(full_dataset_defaults_paydex_ratin_two_years_before))



#vizualize NA's
vis_miss(full_dataset_defaults_paydex_rating) + theme(axis.text.x = element_text(an-
gle=90))
gg_miss_upset(full_dataset_defaults_paydex_rating)

vis_miss(full_dataset_defaults_paydex_ratin_one_year_before) + theme(axis.text.x = ele-
ment_text(angle=90))
gg_miss_upset(full_dataset_defaults_paydex_rating)

#random forest imputation

#str(full_dataset_defaults_paydex_rating)

full_dataset_defaults_paydex_rating <- as.data.frame(full_dataset_defaults_paydex_rat-
ing)

str(full_dataset_defaults_paydex_rating)
ncol(full_dataset_defaults_paydex_rating)

#set amount of cores
registerDoParallel(cores = 6)
```

```
full_dataset_defaults_paydex_rating_imp_forest_no_variablevise <- missForest(full_da-
taset_defaults_paydex_rat-
ing[c(3,4,5,6,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33)],  ver-
bose = T, parallelize = "variables")
full_dataset_defaults_paydex_rating_imp_forest_no_variablevise$OOBerror

#Replace imputed columns in original data frame
full_dataset_defaults_paydex_rating_imputed <- full_dataset_defaults_paydex_rating

full_dataset_defaults_paydex_rating_imp_forest_no_variablevise.imp <- full_dataset_de-
faults_paydex_rating_imp_forest_no_variablevise$ximp

full_dataset_defaults_paydex_rating_imp_forest_no_variablevise.imputed <-replace_col-
umns(full_dataset_defaults_paydex_rating_imputed, full_dataset_defaults_paydex_rat-
ing_imp_forest_no_variablevise.imp, add.unique = TRUE)

str(full_dataset_defaults_paydex_rating_imp_forest_no_variablevise.imputed)

full_dataset_defaults_paydex_rating_imputed <- full_dataset_defaults_paydex_rat-
ing_imp_forest_no_variablevise.imputed


#Calculate additional columns

full_dataset_defaults_paydex_rating_imputed <- full_dataset_defaults_paydex_rating_im-
puted %>%
  mutate(Default_to_net_sales_ratio = ifelse(Turnover == 0, 0, Average_amount_of_de-
faults/Turnover)) %>%
  mutate(Operational_result_to_turnover = ifelse(Turnover == 0, 0,  Operating_re-
sult/Turnover)) %>%
  mutate(Net_result_to_turnover = ifelse(Turnover == 0, 0, Net_result/Turnover)) %>%
  mutate(Fiscal_result_to_turnover = ifelse(Turnover == 0, 0,  Fiscal_year_result/Turno-
ver)) %>%
  mutate(Change_of_turnover_precentage = ifelse(Turnover == 0, 0, Change_of_turno-
ver/Turnover))

#splits the full_dataset to years x before default.

one_year_before <- full_dataset_defaults_paydex_rating_imputed[with(full_dataset_de-
faults_paydex_rating_imputed, (Registered_months_before_defaulting  >= 1 & Regis-
tered_months_before_defaulting  <= 23)), ]

two_years_before <- full_dataset_defaults_paydex_rating_imputed[with(full_dataset_de-
faults_paydex_rating_imputed, (Registered_months_before_defaulting  >= 24 & Regis-
tered_months_before_defaulting  <= 35)), ]
 ]
more_than_two_years <- full_dataset_defaults_paydex_rating_imputed[with(full_dataset_de-
faults_paydex_rating_imputed, (Registered_months_before_defaulting >= 36 & Regis-
tered_months_before_defaulting  <= 100)), ]
```

9.2.3 Split to training & test set

```
library(caret)
library(dplyr)
library(tidyverse)

set.seed(567)
# Store row numbers for training set: index_train
index_train <- createDataPartition(y = one_year_before$Default_status,
                                   p = .7,
                                   list = FALSE)

# Create training set: training_set
training_set <- one_year_before[index_train, ]

# Create test set: test_set
test_set <- one_year_before[-index_train, ]
```

9.2.4 Multivariate Discriminant Analysis MDA

```
library(ROCR, warn.conflicts = FALSE)
library(pROC, warn.conflicts = FALSE)
library(caret)
library(mda)
library(ggplot2)
library(dplyr)
library(tidyverse)

#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              summaryFunction = twoClassSummary, #lisätty testaamatta
                              classProbs = TRUE,
                              savePredictions = "all",
                              sampling = "up")


# training the model (final model)
set.seed(567)
model_mda <- train(factor(Default_status)~
                     Rating
                   +Average_paydex
                   +Number_of_defaults
                   +Average_amount_of_defaults
                   #+Legalform
                   +Turnover
                   +Change_of_turnover
                   +Employee_costs_12Months
                   #+Operating_result
                   #+Net_result
                   +Financing_result
                   +Fiscal_year_result
                   +Return_on_total_assets
                   +Return_on_investments
                   +Return_on_equity
                   #+Equity
                   +Equity_ratio
                   +Debt_to_net_sales_ratio
                   +Gearing
                   +Quick_ratio
                   +Current_ratio
                   +Days_of_receivable_outstanding
                   +Days_of_payables_outstanding
                   +Age_in_months
                   +Industry
                   +Default_to_net_sales_ratio
                   #+Operational_result_to_turnover
                   #+Net_result_to_turnover
                   #+Fiscal_result_to_turnover
                   +Change_of_turnover_precentage,
                   data=training_set,
                   preProcess=c("center","scale"),
                   method="pda2",
                   trControl=train_control,
                   metric = "ROC")

model_mda


#whichTwoPct <- tolerance(model_mda$results, metric = "ROC",
                      #  tol = 2, maximize = TRUE)
```

```
#model_mda_df2 <- update(model_mda, list(df = 2))



# Relative variable importance
library(ggplot2)
ggplot(varImp(model_mda), top = 22)


# Make predictions class outputs
predictions_model_mda <- predict(model_mda , newdata = test_set, type = "raw")
summary(predictions_model_mda)

confusionMatrix(predictions_model_mda, test_set$Default_status)

# Make predictions with propabilities
predictions_model_mda_prob <- predict(model_mda,  newdata = test_set, type = "prob")


# Model performance metrics
data.frame(
  RMSE = RMSE(as.numeric(predictions_model_mda_prob[,2]), test_set$Default_status_prob),
  Rsquare = R2(as.numeric(predictions_model_mda_prob[,2]), test_set$Default_status_prob)
)

#ROC AUC
pred_mda <- prediction(as.numeric(predictions_model_mda_prob[,2]), as.nu-
meric(test_set$Default_status_prob))
perf_mda <- performance(pred_mda , "tpr", "fpr")
plot(perf_mda , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="MDA - ROC curve one
year before", add=FALSE)


AUC_mda <- as.numeric(performance(pred_mda , "auc")@y.values)
AUC_mda

#Add auc to plot
text(0.5,0.4, "AUC = 0.91", font=2)

#PR AUC
perf_PR_mda  <- performance(pred_mda, "prec", "rec")
plot(perf_PR_mda  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="MDA - PR curve one
year before", add=FALSE)

PR_AUC_mda <- as.numeric(performance(pred_mda, "aucpr")@y.values)
PR_AUC_mda

#Add auc to plot
text(0.5,0.4, "AUC = 0.79", font=2)

#calculate brier score
brierScore_mda <- mean((as.numeric(predictions_model_mda_prob[,2])-test_set$Default_sta-
tus_prob)^2)
brierScore_mda

#two years before

predictions_model_mda_prob_two <- predict(model_mda,  newdata = two_years_before, type =
"prob")

#ROC AUC
pred_mda_two <- prediction(as.numeric(predictions_model_mda_prob_two[,2]), as.nu-
meric(two_years_before$Default_status_prob))
perf_mda_two <- performance(pred_mda_two , "tpr", "fpr")
plot(perf_mda_two , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="MDA - ROC curve two
years before", add=FALSE)
```

```
AUC_mda_two <- as.numeric(performance(pred_mda_two , "auc")@y.values)
AUC_mda_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.85", font=2)

#PR AUC
perf_PR_mda_two  <- performance(pred_mda_two, "prec", "rec")
plot(perf_PR_mda_two  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="MDA - PR curve one
year before", add=FALSE)

PR_AUC_mda_two <- as.numeric(performance(pred_mda_two, "aucpr")@y.values)
PR_AUC_mda_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.52", font=2)

#calculate brier score
brierScore_mda_two <- mean((as.numeric(predictions_model_mda_prob_two[,2])-two_years_be-
fore$Default_status_prob)^2)
brierScore_mda_two
```

9.2.5 Stepwise Logistic Regression

```
library(ROCR)
library(pROC)
library(MASS)
library(caret)
library(dplyr)
library(tidyverse)
library(ggplot2)


#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              savePredictions = TRUE,
                              classProbs = TRUE,
                              verboseIter = TRUE,
                              summaryFunction = twoClassSummary,
                              sampling = "up")

# building the model
set.seed(567)
stepwise_logit <- train(factor(Default_status) ~
              Rating
            +Average_paydex
            +Number_of_defaults
            +Average_amount_of_defaults
            +Legalform
            +Turnover
            +Change_of_turnover
            +Employee_costs_12Months
            +Operating_result
            +Net_result
            +Financing_result
            +Fiscal_year_result
            +Return_on_total_assets
            +Return_on_investments
            +Return_on_equity
            +Equity
```

```
            +Equity_ratio
            +Debt_to_net_sales_ratio
            +Gearing
            +Quick_ratio
            +Current_ratio
            +Days_of_receivable_outstanding
            +Days_of_payables_outstanding
            +Age_in_months
            +Industry
            +Default_to_net_sales_ratio
            +Operational_result_to_turnover
            +Net_result_to_turnover
            +Fiscal_result_to_turnover
            +Change_of_turnover_precentage,
            data = training_set,
            trControl = train_control,
            method = "glmStepAIC",
            family="binomial", # Logistic regression is specified,
            direction = "backward",
            preProcess=c("center","scale"),
            metric = "ROC"
)

print(stepwise_logit)
stepwise_logit

warnings(stepwise_logit)

summary(stepwise_logit$finalModel)


#  model coefficients
coef(stepwise_logit, stepwise_logit$bestTune)


#visualise the most improtant variables
library(ggplot2)
ggplot(varImp(stepwise_logit))

#most important variables
stepwise_logit_importance <- varImp(stepwise_logit)

ggplot(stepwise_logit_importance, top = 30)

predictors(stepwise_logit)

# Make predictions for the class
predictions_glm <- predict(stepwise_logit, newdata = test_set, type = "raw")

head(predictions_glm)

confusionMatrix(predictions_glm, test_set$Default_status)


#Predicts the probabilities and not the class
predictions_glm_prob <- predict(stepwise_logit, newdata = test_set, type = "prob")

#ROC AUC

predROC_logit<- prediction(as.numeric(predictions_glm_prob[,2]), as.numeric(test_set$De-
fault_status_prob))
perf_ROC_logit<- performance(predROC_logit, "tpr", "fpr")
plot(perf_ROC_logit, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Stepwise logistic re-
gression - ROC curve one year before", add=FALSE)


AUC_stepwise_logit <- as.numeric(performance(predROC_logit , "auc")@y.values)
AUC_stepwise_logit


#Add auc to plot
```

```
text(0.5,0.4, "AUC = 0.92", font=2)

#PR AUC

perf_PR_logit  <- performance(predROC_logit, "prec", "rec")
plot(perf_PR_logit, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Stepwise logistic re-
gression - PR curve one year before", add=FALSE)

PR_AUC_stepwise_logit <- as.numeric(performance(predROC_logit, "aucpr")@y.values)
PR_AUC_stepwise_logit

#Add auc to plot
text(0.5,0.5, "AUC = 0.86", font=2)

#calculate brier score
brierScore_stepwise_logit <- mean((as.numeric(predictions_glm_prob[,2])-test_set$De-
fault_status_prob)^2)
brierScore_stepwise_logit


#predict with two years before data
predictions_glm_prob_two <- predict(stepwise_logit, newdata = two_years_before, type =
"prob")

#ROC AUC

predROC_logit_two<- prediction(as.numeric(predictions_glm_prob_two[,2]), as.nu-
meric(two_years_before$Default_status_prob))
perf_ROC_logit_two<- performance(predROC_logit_two, "tpr", "fpr")
plot(perf_ROC_logit_two, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Stepwise logistic re-
gression - ROC curve two years before", add=FALSE)


AUC_stepwise_logit_two <- as.numeric(performance(predROC_logit_two, "auc")@y.values)
AUC_stepwise_logit_two


#Add auc to plot
text(0.5,0.4, "AUC = 0.89", font=2)

#PR AUC

perf_PR_logit  <- performance(predROC_logit_two, "prec", "rec")
plot(perf_PR_logit , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Stepwise logistic re-
gression - PR curve two years before", add=FALSE)

PR_AUC_stepwise_logit_two <- as.numeric(performance(predROC_logit_two, "aucpr")@y.val-
ues)
PR_AUC_stepwise_logit_two

text(0.5,0.5, "AUC = 0.70", font=2)

#calculate brier score
brierScore_stepwise_logit <- mean((as.numeric(predictions_glm_prob_two[,2])-
two_years_before$Default_status_prob)^2)
brierScore_stepwise_logit
```

9.2.6 LASSO Logistic Regression

```
library(ROCR)
library(pROC)
library(glmnet)
library(caret)
library(ggplot2)
library(dplyr)
library(tidyverse)
```

```
#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              summaryFunction = twoClassSummary, #lisätty testaamatta
                              classProbs = TRUE,
                              savePredictions = "all",
                              sampling = "up")

# Lamda vector
lambda_vector <-10^seq(10, -2, length = 100)


# training the model
set.seed(567)
logit_lasso <- train(factor(Default_status)~
                       Rating
                     +Average_paydex
                     +Number_of_defaults
                     +Average_amount_of_defaults
                     +Legalform
                     +Turnover
                     +Change_of_turnover
                     +Employee_costs_12Months
                     +Operating_result
                     +Net_result
                     +Financing_result
                     +Fiscal_year_result
                     +Return_on_total_assets
                     +Return_on_investments
                     +Return_on_equity
                     +Equity
                     +Equity_ratio
                     +Debt_to_net_sales_ratio
                     +Gearing
                     +Quick_ratio
                     +Current_ratio
                     +Days_of_receivable_outstanding
                     +Days_of_payables_outstanding
                     +Age_in_months
                     +Industry
                     +Default_to_net_sales_ratio
                     +Operational_result_to_turnover
                     +Net_result_to_turnover
                     +Fiscal_result_to_turnover
                     +Change_of_turnover_precentage,
                    data=training_set,
                    preProcess=c("center","scale"),
                    method="glmnet",
                    metric = "ROC",
                    family="binomial", # Logistic regression is specified,
                    tuneGrid=expand.grid(alpha=1, lambda=lambda_vector),
                    trControl=train_control)

logit_lasso

# Best tuning parameters (alpha, lambda)
logit_lasso$bestTune

# Relative variable importance
library(ggplot2)
ggplot(varImp(logit_lasso))

logit_lasso_importance <- varImp(logit_lasso)

ggplot(logit_lasso_importance, top = 21)
```

```
# model coefficients
coef(logit_lasso$finalModel, logit_lasso$bestTune$lambda)

predictors(logit_lasso)

# Make predictions with class
predictions_logit_lasso_class <- predict(logit_lasso, newdata = test_set, type = "raw")
summary(predictions_logit_lasso_class)

confusionMatrix(predictions_logit_lasso_class, test_set$Default_status)

# Make predictions with probabilities
predictions_logit_lasso <- predict(logit_lasso, newdata = test_set, type = "prob")
str(predictions_logit_lasso)

# Model performance metrics
data.frame(
  RMSE = RMSE(as.numeric(predictions_logit_lasso[,2]), test_set$Default_status_prob),
  Rsquare = R2(as.numeric(predictions_logit_lasso[,2]), test_set$Default_status_prob)
)

#ROC AUC
predROC_logit_lasso<- prediction(as.numeric(predictions_logit_lasso[,2]), as.nu-
meric(test_set$Default_status_prob))
perf_ROC_logit_lasso <- performance(predROC_logit_lasso, "tpr", "fpr")
plot(perf_ROC_logit_lasso, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="LASSO logistic re-
gression - ROC curve one year before", add=FALSE)


AUC_logit_lasso <- as.numeric(performance(predROC_logit_lasso , "auc")@y.values)
AUC_logit_lasso

#Add auc to plot
text(0.5,0.5, "AUC = 0.91", font=2)

#PR AUC

perf_PR_logit_lasso  <- performance(predROC_logit_lasso, "prec", "rec")
plot(perf_PR_logit_lasso , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="LASSO logistic re-
gression - PR curve one year before", add=FALSE)

PR_AUC_logit_lasso <- as.numeric(performance(predROC_logit_lasso, "aucpr")@y.values)
PR_AUC_logit_lasso

#Add auc to plot
text(0.5,0.4, "AUC = 0.83", font=2)


#calculate brier score
brierScore_logit_lasso <- mean((as.numeric(predictions_logit_lasso[,2])-test_set$De-
fault_status_prob)^2)
brierScore_logit_lasso

#predict with two years before data

predictions_logit_lasso_two <- predict(logit_lasso, newdata = two_years_before, type =
"prob")

#ROC AUC
predROC_logit_lasso_two<- prediction(as.numeric(predictions_logit_lasso_two[,2]), as.nu-
meric(two_years_before$Default_status_prob))
perf_ROC_logit_lasso_two <- performance(predROC_logit_lasso_two, "tpr", "fpr")
plot(perf_ROC_logit_lasso_two, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="LASSO logistic re-
gression - ROC curve two years before", add=FALSE)


AUC_logit_lasso_two <- as.numeric(performance(predROC_logit_lasso_two, "auc")@y.values)
AUC_logit_lasso_two
```

```
#Add auc to plot
text(0.5,0.4, "AUC = 0.86", font=2)

#PR AUC

perf_PR_logit_lasso_two  <- performance(predROC_logit_lasso_two, "prec", "rec")
plot(perf_PR_logit_lasso_two , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7),  main="LASSO logistic re-
gression - PR curve two years before", add=FALSE)

PR_AUC_logit_lasso_two <- as.numeric(performance(predROC_logit_lasso_two,
"aucpr")@y.values)
PR_AUC_logit_lasso_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.58", font=2)


#calculate brier score
brierScore_logit_lasso <- mean((as.numeric(predictions_logit_lasso_two[,2])-
two_years_before$Default_status_prob)^2)
brierScore_logit_lasso
```

9.2.7 Decision Tree CART

```
library(ROCR)
library(pROC)
library(caret)
library(rpart, warn.conflicts = FALSE)
library(rpart.plot, warn.conflicts = FALSE)
library(rattle, warn.conflicts = FALSE)
library(RColorBrewer, warn.conflicts = FALSE)
library(ggplot2)
library(dplyr)
library(tidyverse)


#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              summaryFunction = twoClassSummary, #lisätty testaamatta
                              classProbs = TRUE,
                              savePredictions = "all",
                              sampling = "up")

tunegrid <- expand.grid(.cp=seq(0,1,by=0.01))


# training the model
set.seed(567)
model_cart <- train(factor(Default_status)~
                      Rating
                    +Average_paydex
                    +Number_of_defaults
                    +Average_amount_of_defaults
                    +Legalform
                    +Turnover
                    +Change_of_turnover
                    +Employee_costs_12Months
                    +Operating_result
                    +Net_result
                    +Financing_result
                    +Fiscal_year_result
```

```
                    +Return_on_total_assets
                    +Return_on_investments
                    +Return_on_equity
                    +Equity
                    +Equity_ratio
                    +Debt_to_net_sales_ratio
                    +Gearing
                    +Quick_ratio
                    +Current_ratio
                    +Days_of_receivable_outstanding
                    +Days_of_payables_outstanding
                    +Age_in_months
                    +Industry
                    +Default_to_net_sales_ratio
                    +Operational_result_to_turnover
                    +Net_result_to_turnover
                    +Fiscal_result_to_turnover
                    +Change_of_turnover_precentage,
                    data=training_set,
                    #preProcess=c("center","scale"),
                    method="rpart",
                    trControl=train_control,
                    tuneGrid=tunegrid)

model_cart

# Relative variable importance
library(ggplot2)
ggplot(varImp(model_cart), top = 16)


#ggplot(model_cart, top = 30)


#visualize the tree

prp(model_cart$finalModel, type = 5, varlen = 0, yesno = 0,  yes.text="default",
no.text="no default")

# Make predictions
predictions_model_cart <- predict(model_cart , newdata = test_set, type = "raw")
summary(predictions_model_cart)

confusionMatrix(predictions_model_cart, test_set$Default_status)

# Make predictions for all models using the test set
predictions_model_cart_prob <- predict(model_cart,  newdata = test_set, type = "prob")


# Model performance metrics
data.frame(
  RMSE = RMSE(as.numeric(predictions_model_cart_prob[,2]), test_set$Default_sta-
tus_prob),
  Rsquare = R2(as.numeric(predictions_model_cart_prob[,2]), test_set$Default_sta-
tus_prob)
)

#ROC AUC
pred_cart <- prediction(as.numeric(predictions_model_cart_prob[,2]), as.nu-
meric(test_set$Default_status_prob))
perf_cart <- performance(pred_cart , "tpr", "fpr")
plot(perf_cart , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="CART - ROC curve one
year before", add=FALSE)


AUC_cart <- as.numeric(performance(pred_cart , "auc")@y.values)
AUC_cart

#Add auc to plot
text(0.5,0.4, "AUC = 0.90", font=2)
```

```
#PR AUC
perf_PR_cart  <- performance(pred_cart, "prec", "rec")
plot(perf_PR_cart  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="CART - PR curve one
year before", add=FALSE)

PR_AUC_cart <- as.numeric(performance(pred_cart, "aucpr")@y.values)
PR_AUC_cart

#Add auc to plot
text(0.8,0.5, "AUC = 0.75", font=2)

#calculate brier score
brierScore_cart <- mean((as.numeric(predictions_model_cart_prob[,2])-test_set$De-
fault_status_prob)^2)
brierScore_cart

#two years before

predictions_model_cart_prob_two <- predict(model_cart,  newdata = two_years_before, type
= "prob")

#ROC AUC
pred_cart_two <- prediction(as.numeric(predictions_model_cart_prob_two[,2]), as.nu-
meric(two_years_before$Default_status_prob))
perf_cart_two <- performance(pred_cart_two , "tpr", "fpr")
plot(perf_cart_two , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="CART - ROC curve two
years before", add=FALSE)


AUC_cart_two <- as.numeric(performance(pred_cart_two , "auc")@y.values)
AUC_cart_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.85", font=2)

#PR AUC
perf_PR_cart_two  <- performance(pred_cart_two, "prec", "rec")
plot(perf_PR_cart_two  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="CART - PR curve two
years before", add=FALSE)

PR_AUC_cart_two <- as.numeric(performance(pred_cart_two, "aucpr")@y.values)
PR_AUC_cart_two

#Add auc to plot
text(0.7,0.35, "AUC = 0.52", font=2)

#calculate brier score
brierScore_cart_two <- mean((as.numeric(predictions_model_cart_prob_two[,2])-
two_years_before$Default_status_prob)^2)
brierScore_cart_two
```

9.2.8 Random Forest

```
library(ROCR)
library(pROC)
library(MASS)
library(caret)
library(ggplot2)
library(dplyr)
library(tidyverse)

#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
```

```
                    method = 'cv',
                    number = folds,
                    savePredictions = TRUE,
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE,
                    sampling = "up")

tuneGrid<- expand.grid(.mtry=c(1:15))

# training the model

set.seed(567)
rf_10_k_fold <- train(factor(Default_status)~
                Rating
                +Average_paydex
                +Number_of_defaults
                +Average_amount_of_defaults
                +Legalform
                +Turnover
                +Change_of_turnover
                +Employee_costs_12Months
                +Operating_result
                +Net_result
                +Financing_result
                +Fiscal_year_result
                +Return_on_total_assets
                +Return_on_investments
                +Return_on_equity
                +Equity
                +Equity_ratio
                +Debt_to_net_sales_ratio
                +Gearing
                +Quick_ratio
                +Current_ratio
                +Days_of_receivable_outstanding
                +Days_of_payables_outstanding
                +Age_in_months
                +Industry
                +Default_to_net_sales_ratio
                +Operational_result_to_turnover
                +Net_result_to_turnover
                +Fiscal_result_to_turnover
                +Change_of_turnover_precentage,
                data = training_set,
                trControl = train_control,
                method = "rf",
                metric = "ROC",
                tuneGrid=tuneGrid
                )

print(rf_10_k_fold)

#predict and confusion matrix with classification / major vote
pred_rf_class <- predict(rf_10_k_fold, newdata = test_set, type="raw")
head(pred_rf_class)

confusionMatrix(pred_rf_class, test_set$Default_status)

#most important variables
rf_importance <- varImp(rf_10_k_fold)

ggplot(rf_importance, top = 20)

#predict with test set
predictions_rf_prob <- predict(rf_10_k_fold, newdata = test_set, type="prob",
norm.votes=TRUE, predict.all=FALSE, proximity=FALSE, nodes=FALSE)

#ROC AUC
predROC_rf<- prediction(as.numeric(predictions_rf_prob[,2]), as.numeric(test_set$De-
fault_status))
perf_ROC_rf <- performance(predROC_rf, "tpr", "fpr")
plot(perf_ROC_rf, colorize = TRUE,
```

```
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Random Forest - ROC
curve one year before", add=FALSE)


AUC_rf <- as.numeric(performance(predROC_rf , "auc")@y.values)
AUC_rf

#Add auc to plot
text(0.5,0.5, "AUC = 0.97", font=2)

#PR AUC

perf_PR_rf  <- performance(predROC_rf, "prec", "rec")
plot(perf_PR_rf, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Random Forest - PR
curve one year before", add=FALSE)

PR_AUC_rf <- as.numeric(performance(predROC_rf , "aucpr")@y.values)
PR_AUC_rf

#Add auc to plot
text(0.5,0.5, "AUC = 0.93", font=2)


#calculate brier score
brierScore_pred_rf <- mean((as.numeric(predictions_rf_prob[,2])-test_set$Default_sta-
tus_prob)^2)
brierScore_pred_rf


#predict with two years before data
predictions_rf_prob_two <- predict(rf_10_k_fold, newdata = two_years_before,
type="prob", norm.votes=TRUE, predict.all=FALSE, proximity=FALSE, nodes=FALSE)

#AUC ROC
predROC_rf_two<- prediction(as.numeric(predictions_rf_prob_two[,2]), as.nu-
meric(two_years_before$Default_status))
perf_ROC_rf_two <- performance(predROC_rf_two, "tpr", "fpr")
plot(perf_ROC_rf_two, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Random Forest - ROC
curve two years before", add=FALSE)



AUC_rf_two <- as.numeric(performance(predROC_rf_two , "auc")@y.values)
AUC_rf_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.93", font=2)

#PR AUC

perf_PR_rf_two  <- performance(predROC_rf_two, "prec", "rec")
plot(perf_PR_rf_two, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Random Forest - PR
curve two years before", add=FALSE)

PR_AUC_rf <- as.numeric(performance(predROC_rf , "aucpr")@y.values)
PR_AUC_rf

#Add auc to plot
text(0.5,0.5, "AUC = 0.93", font=2)

#calculate brier score two years before data
brierScore_pred_rf_two <- mean((as.numeric(predictions_rf_prob_two[,2])-two_years_be-
fore$Default_status_prob)^2)
brierScore_pred_rf_two
```

9.2.9 AdaBoost

```
library(ROCR)
library(pROC)
library(caret)
library(fastAdaboost)
library(rpart)
library(rpart.plot)
library(rattle)
library(ggplot2)
library(dplyr)
library(tidyverse)

#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              summaryFunction = twoClassSummary, #lisätty testaamatta
                              classProbs = TRUE,
                              savePredictions = "all",
                              sampling = "up")

tunegrid <- expand.grid(.cp=seq(0,1,by=0.01))


# training the model
set.seed(567)
model_ada  <- train(factor(Default_status)~
                      Rating
                    +Average_paydex
                    +Number_of_defaults
                    +Average_amount_of_defaults
                    +Legalform
                    +Turnover
                    +Change_of_turnover
                    +Employee_costs_12Months
                    +Operating_result
                    +Net_result
                    +Financing_result
                    +Fiscal_year_result
                    +Return_on_total_assets
                    +Return_on_investments
                    +Return_on_equity
                    +Equity
                    +Equity_ratio
                    +Debt_to_net_sales_ratio
                    +Gearing
                    +Quick_ratio
                    +Current_ratio
                    +Days_of_receivable_outstanding
                    +Days_of_payables_outstanding
                    +Age_in_months
                    +Industry
                    +Default_to_net_sales_ratio
                    +Operational_result_to_turnover
                    +Net_result_to_turnover
                    +Fiscal_result_to_turnover
                    +Change_of_turnover_precentage,
                    data=training_set,
                    #preProcess=c("center","scale"),
                    method="adaboost",
                    #tuneGrid=tunegrid,
                    metric = "ROC",
                    trControl=train_control
                    )

model_ada
```

```
# Relative variable importance
library(ggplot2)
ggplot(varImp(model_ada), top = 29)



#visualize the tree

# Make predictions
predictions_model_ada <- predict(model_ada , newdata = test_set, type = "raw")
summary(predictions_model_ada)

confusionMatrix(predictions_model_ada, test_set$Default_status)

# Make predictions for all models using the test set
predictions_model_ada_prob <- predict(model_ada,  newdata = test_set, type = "prob")


# Model performance metrics
data.frame(
  RMSE = RMSE(as.numeric(predictions_model_ada_prob[,2]), test_set$Default_status_prob),
  Rsquare = R2(as.numeric(predictions_model_ada_prob[,2]), test_set$Default_status_prob)
)

#ROC AUC
pred_ada <- prediction(as.numeric(predictions_model_ada_prob[,2]), as.nu-
meric(test_set$Default_status_prob))
perf_ada <- performance(pred_ada , "tpr", "fpr")
plot(perf_ada , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="AdaBoost - ROC curve
one year before", add=FALSE)


AUC_ada <- as.numeric(performance(pred_ada , "auc")@y.values)
AUC_ada

#Add auc to plot
text(0.5,0.4, "AUC = 0.98", font=2)

#PR AUC
perf_PR_ada  <- performance(pred_ada, "prec", "rec")
plot(perf_PR_ada  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="AdaBoost - PR curve
one year before", add=FALSE)

PR_AUC_ada <- as.numeric(performance(pred_ada, "aucpr")@y.values)
PR_AUC_ada

#Add auc to plot
text(0.5,0.6, "AUC = 0.95", font=2)

#calculate brier score
brierScore_ada <- mean((as.numeric(predictions_model_ada_prob[,2])-test_set$Default_sta-
tus_prob)^2)
brierScore_ada

#two years before

predictions_model_ada_prob_two <- predict(model_ada,  newdata = two_years_before, type =
"prob")

#ROC AUC
pred_ada_two <- prediction(as.numeric(predictions_model_ada_prob_two[,2]), as.nu-
meric(two_years_before$Default_status_prob))
perf_ada_two <- performance(pred_ada_two , "tpr", "fpr")
plot(perf_ada_two , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="AdaBoost - ROC curve
two years before", add=FALSE)


AUC_ada_two <- as.numeric(performance(pred_ada_two , "auc")@y.values)
```

```
AUC_ada_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.95", font=2)

#PR AUC
perf_PR_ada_two  <- performance(pred_ada_two, "prec", "rec")
plot(perf_PR_ada_two  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="AdaBoost - PR curve
two years before", add=FALSE)

PR_AUC_ada_two <- as.numeric(performance(pred_ada_two, "aucpr")@y.values)
PR_AUC_ada_two

#Add auc to plot
text(0.5,0.5, "AUC = 0.86", font=2)

#calculate brier score
brierScore_ada_two <- mean((as.numeric(predictions_model_ada_prob_two[,2])-two_years_be-
fore$Default_status_prob)^2)
brierScore_ada_two
```

9.2.10 Support Vector Machine

```
#Computing SVM using polynomial basis kernel:

library(ROCR)
library(pROC
#library(e1071)
library(caret)
library(kernlab) #used for radial model



#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              summaryFunction = twoClassSummary,
                              classProbs = TRUE,
                              savePredictions = "all",
                              sampling = "up")



# building the model
set.seed(567)
model_svm_poly <- train(factor(Default_status)~
                          Rating
                        +Average_paydex
                        +Number_of_defaults
                        +Average_amount_of_defaults
                        +Legalform
                        +Turnover
                        +Change_of_turnover
                        +Employee_costs_12Months
                        +Operating_result
                        +Net_result
                        +Financing_result
                        +Fiscal_year_result
                        +Return_on_total_assets
                        +Return_on_investments
                        +Return_on_equity
                        +Equity
                        +Equity_ratio
```

```
                            +Debt_to_net_sales_ratio
                            +Gearing
                            +Quick_ratio
                            +Current_ratio
                            +Days_of_receivable_outstanding
                            +Days_of_payables_outstanding
                            +Age_in_months
                            +Industry
                            +Default_to_net_sales_ratio
                            +Operational_result_to_turnover
                            +Net_result_to_turnover
                            +Fiscal_result_to_turnover
                            +Change_of_turnover_precentage,
                            data=training_set,
                            preProcess=c("center","scale"),
                            method="svmPoly",
                            trControl=train_control,
                            metric = "ROC",
                            tuneLength = 4
                          )

model_svm_poly
plot(model_svm_poly)

# Best tuning parameters (alpha, lambda)
model_svm_poly$bestTune

# Visualize the importance of different predictor variables
library(ggplot2)
ggplot(varImp(model_svm_poly))


# Make predictions
predictions_model_svm_poly <- predict(model_svm_poly , newdata = test_set, type = "raw")
summary(predictions_model_svm_poly)

confusionMatrix(predictions_model_svm_poly, test_set$Default_status)

# Make predictions for all models using the test set
predictions_model_svm_poly_prob <- predict(model_svm_poly,  newdata = test_set, type =
"prob")


# Model performance metrics
data.frame(
  RMSE = RMSE(as.numeric(predictions_model_svm_poly_prob[,2]), test_set$Default_sta-
tus_prob),
  Rsquare = R2(as.numeric(predictions_model_svm_poly_prob [,2]), test_set$Default_sta-
tus_prob)
)

#ROC AUC
pred_svm_poly <- prediction(as.numeric(predictions_model_svm_poly_prob[,2]), as.nu-
meric(test_set$Default_status_prob))
perf_svm_poly <- performance(pred_svm_poly , "tpr", "fpr")
plot(perf_svm_poly , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="SVM - ROC curve one
year before", add=FALSE)


AUC_svm_poly <- as.numeric(performance(pred_svm_poly , "auc")@y.values)
AUC_svm_poly

#Add auc to plot
text(0.5,0.4, "AUC = 0.91", font=2)

#PR AUC
perf_PR_svm_poly  <- performance(pred_svm_poly, "prec", "rec")
plot(perf_PR_svm_poly  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="SVM - PR curve one
year before", add=FALSE)
```

```
PR_AUC_svm_poly <- as.numeric(performance(pred_svm_poly, "aucpr")@y.values)
PR_AUC_svm_poly

#Add auc to plot
text(0.5,0.5, "AUC = 0.83", font=2)

#calculate brier score
brierScore_svm_poly <- mean((as.numeric(predictions_model_svm_poly_prob[,2])-
test_set$Default_status_prob)^2)
brierScore_svm_poly



#two years before

predictions_model_svm_poly_prob_two <- predict(model_svm_poly,  newdata = two_years_be-
fore, type = "prob")

#ROC AUC
pred_svm_poly_two <- prediction(as.numeric(predictions_model_svm_poly_prob_two[,2]),
as.numeric(two_years_before$Default_status_prob))
perf_svm_poly_two <- performance(pred_svm_poly_two , "tpr", "fpr")
plot(perf_svm_poly_two , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7),  main="SVM - ROC curve two
years before", add=FALSE)


AUC_svm_poly_two <- as.numeric(performance(pred_svm_poly_two , "auc")@y.values)
AUC_svm_poly_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.88", font=2)

#PR AUC
perf_PR_svm_polytwo  <- performance(pred_svm_poly_two, "prec", "rec")

plot(perf_PR_svm_polytwo, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7),  main="SVM - PR curve two
years before", add=FALSE)


PR_AUC_svm_poly_two <- as.numeric(performance(pred_svm_poly_two , "aucpr")@y.values)
PR_AUC_svm_poly_two

#Add auc to plot
text(0.5,0.5, "AUC = 0.83", font=2)


#calculate brier score
brierScore_svm_poly_two <- mean((as.numeric(predictions_model_svm_poly_prob_two[,2])-
two_years_before$Default_status_prob)^2)
brierScore_svm_poly_two
```

9.2.11 Neural Network

```
library(ROCR)
library(pROC)
library(caret)
library(nnet)
library(ggplot2)
library(dplyr)
library(tidyverse)
```

```
#k 10 fold  cross validation

folds <- 10
cvIndex <- createFolds(factor(training_set$Default_status), folds, returnTrain = T)
train_control <- trainControl(index = cvIndex,
                              method = 'cv',
                              number = folds,
                              summaryFunction = twoClassSummary, #lisätty testaamatta
                              classProbs = TRUE,
                              #sampling = "up",
                              savePredictions = "all",
                              #linout = TRUE,
                              )


tuneGrid<- expand.grid(size = seq(from = 1, to = 10, by = 0.5),
                       decay = seq(from = 0.1, to = 0.5, by = 0.1))

# training the model
set.seed(567)
model_nnet <- train(factor(Default_status)~
                     Rating
                    +Average_paydex
                    +Number_of_defaults
                    +Average_amount_of_defaults
                    +Legalform
                    +Turnover
                    +Change_of_turnover
                    +Employee_costs_12Months
                    +Operating_result
                    +Net_result
                    +Financing_result
                    +Fiscal_year_result
                    +Return_on_total_assets
                    +Return_on_investments
                    +Return_on_equity
                    +Equity
                    +Equity_ratio
                    +Debt_to_net_sales_ratio
                    +Gearing
                    +Quick_ratio
                    +Current_ratio
                    +Days_of_receivable_outstanding
                    +Days_of_payables_outstanding
                    +Age_in_months
                    +Industry
                    +Default_to_net_sales_ratio
                    +Operational_result_to_turnover
                    +Net_result_to_turnover
                    +Fiscal_result_to_turnover
                    +Change_of_turnover_precentage,
                    data=training_set,
                    preProcess=c("center","scale"),
                    method="nnet",
                    trControl=train_control,
                    tuneGrid=tuneGrid,
                    metric = "ROC")

model_nnet



# Relative predictor variable importance
library(ggplot2)
ggplot(varImp(model_nnet), top = 53)


# Make predictions
predictions_model_nnet <- predict(model_nnet , newdata = test_set, type = "raw")
summary(predictions_model_nnet)

confusionMatrix(predictions_model_nnet, test_set$Default_status)
```

```
# Make predictions for all models using the test set
predictions_model_nnet_prob <- predict(model_nnet,  newdata = test_set, type = "prob")


# Model performance metrics
data.frame(
  RMSE = RMSE(as.numeric(predictions_model_nnet_prob[,2]), test_set$Default_sta-
tus_prob),
  Rsquare = R2(as.numeric(predictions_model_nnet_prob[,2]), test_set$Default_sta-
tus_prob)
)

#ROC AUC
pred_nnet <- prediction(as.numeric(predictions_model_nnet_prob[,2]), as.nu-
meric(test_set$Default_status_prob))
perf_nnet <- performance(pred_nnet , "tpr", "fpr")
plot(perf_nnet , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Neural Network - ROC
curve one year before", add=FALSE)


AUC_nnet <- as.numeric(performance(pred_nnet , "auc")@y.values)
AUC_nnet

#Add auc to plot
text(0.5,0.4, "AUC = 0.92", font=2)

#PR AUC
perf_PR_nnet  <- performance(pred_nnet, "prec", "rec")
plot(perf_PR_nnet  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Neural Network - PR
curve one year before", add=FALSE)

PR_AUC_nnet <- as.numeric(performance(pred_nnet, "aucpr")@y.values)
PR_AUC_nnet

#Add auc to plot
text(0.5,0.5, "AUC = 0.87", font=2)

#calculate brier score
brierScore_nnet <- mean((as.numeric(predictions_model_nnet_prob[,2])-test_set$De-
fault_status_prob)^2)
brierScore_nnet

#two years before

predictions_model_nnet_prob_two <- predict(model_nnet,  newdata = two_years_before, type
= "prob")

#ROC AUC
pred_nnet_two <- prediction(as.numeric(predictions_model_nnet_prob_two[,2]), as.nu-
meric(two_years_before$Default_status_prob))
perf_nnet_two <- performance(pred_nnet_two , "tpr", "fpr")
plot(perf_nnet_two , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Neural Network - ROC
curve two years before", add=FALSE)


AUC_nnet_two <- as.numeric(performance(pred_nnet_two , "auc")@y.values)
AUC_nnet_two

#Add auc to plot
text(0.5,0.4, "AUC = 0.89", font=2)

#PR AUC
perf_PR_nnet_two  <- performance(pred_nnet_two, "prec", "rec")
plot(perf_PR_nnet_two  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="Neural Network - PR
curve two years before", add=FALSE)

PR_AUC_nnet_two <- as.numeric(performance(pred_nnet_two, "aucpr")@y.values)
```

```
PR_AUC_nnet_two

#Add auc to plot
text(0.5,0.5, "AUC = 0.68", font=2)

#calculate brier score
brierScore_nnet_two <- mean((as.numeric(predictions_model_nnet_prob_two[,2])-
two_years_before$Default_status_prob)^2)
brierScore_nnet_two
```

9.2.12 External rating metrics

```
library(dplyr)


Bisnode_rating <-read.csv("Bisnode rating.csv", sep=";")

str(Bisnode_rating)

full_dataset_defaults_paydex_rating_imputed_bisnode  <- full_dataset_defaults_pay-
dex_rating_imputed %>%
  left_join(Bisnode_rating,  by = c("Rating"))

str(full_dataset_defaults_paydex_rating_imputed_bisnode) # check the format


write.csv(full_dataset_defaults_paydex_rating_imputed_bisnode, file = "full_dataset_de-
faults_paydex_rating_imputed_bisnode.csv",row.names=FALSE)


one_year_before_bisnode <- full_dataset_defaults_paydex_rating_im-
puted_bisnode[with(full_dataset_defaults_paydex_rating_imputed_bisnode, (Regis-
tered_months_before_defaulting  >= 1 & Registered_months_before_defaulting  <= 23)),
]#%>%
#select(default_status, age_months_in_fiscal_year, equity_ratio, quick_ratio, n_de-
faults, avg_amount_paydex, return_on_total_assets)
two_years_before_bisnode <- full_dataset_defaults_paydex_rating_im-
puted_bisnode[with(full_dataset_defaults_paydex_rating_imputed_bisnode, (Regis-
tered_months_before_defaulting  >= 24 & Registered_months_before_defaulting  <= 35)), ]

#check that amount of companies match
length (unique(one_year_before$Business_id))
length (unique(one_year_before_bisnode$Business_id))
str(one_year_before_bisnode)#check number of variables


#Bisnode prepability of default
Predict_bisnode <- one_year_before_bisnode[,41] #Column 41 bisnode's rating, 42 is in-
terpreted rating

#ROC curve and AUC for the one year before dataset
predROC_Bisnode<- prediction(as.numeric(Predict_bisnode), as.numeric(one_year_be-
fore_bisnode$Default_status))

Bisnode_AUC <- as.numeric(performance(predROC_Bisnode, "auc")@y.values)
Bisnode_AUC

#ROC AUC
perf_ROC_Bisnode <- performance(predROC_Bisnode, "tpr", "fpr")
plot(perf_ROC_Bisnode, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="ROC - 12-month PD one
year before", add=FALSE)
#Add auc to plot
text(0.5,0.4, "AUC = 0.79", font=2)
```

```
#PR AUC
PR_ROC_Bisnode  <- performance(predROC_Bisnode, "prec", "rec")
plot(PR_ROC_Bisnode  , colorize = TRUE,
     print.cutoffs.at = seq(0,0.6,0.1),text.adj = c(0.5,2), main="PR - 12-month PD one
year before", add=FALSE)

Bisnode_PR_AUC <- as.numeric(performance(predROC_Bisnode, "aucpr")@y.values)
Bisnode_PR_AUC

#Add auc to plot
text(0.65,0.4, "AUC = 0.55", font=2)

#brier score

brierScore_Bisnode <- mean((as.numeric(Predict_bisnode)-as.numeric(one_year_be-
fore_bisnode$Default_status_prob))^2)
brierScore_Bisnode


#ROC curve and AUC  for two years before dataset

Predict_bisnode_two <- two_years_before_bisnode[,41]
predROC_Bisnode_two<- prediction(as.numeric(Predict_bisnode_two), as.nu-
meric(two_years_before_bisnode$Default_status))

Bisnode_AUC_two <- as.numeric(performance(predROC_Bisnode_two, "auc")@y.values)
Bisnode_AUC_two

#PROC AUC
perf_ROC_Bisnode_two <- performance(predROC_Bisnode_two, "tpr", "fpr")
plot(perf_ROC_Bisnode_two, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="ROC - 12-month PD two
years before", add=FALSE)
#Add auc to plot
text(0.5,0.4, "AUC = 0,74" , font=2)


#PR AUC
PR_ROC_Bisnode_two  <- performance(predROC_Bisnode_two, "prec", "rec")
plot(PR_ROC_Bisnode_two  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="PR - 12-month PD two
years before", add=FALSE)

Bisnode_PR_AUC_two <- as.numeric(performance(predROC_Bisnode_two, "aucpr")@y.values)
Bisnode_PR_AUC_two

#Add auc to plot
text(0.6,0.2, "AUC = 0.29", font=2)



#brier score

brierScore_Bisnode_two <- mean((as.numeric(Predict_bisnode_two)-as.numeric(two_years_be-
fore_bisnode$Default_status_prob))^2)
brierScore_Bisnode_two



# Predictions for interpreted rating

Predict_bisnode_int <- one_year_before_bisnode[,42] #Column 41 bisnode's rating, 42 is
interpreted rating


#ROC curve and AUC for the one year before dataset
predROC_Bisnode_int<- prediction(as.numeric(Predict_bisnode_int), as.nu-
meric(one_year_before_bisnode$Default_status))

Bisnode_AUC_int <- as.numeric(performance(predROC_Bisnode_int, "auc")@y.values)
```

```
Bisnode_AUC_int

#ROC AUC
perf_ROC_Bisnode_int <- performance(predROC_Bisnode_int, "tpr", "fpr")
plot(perf_ROC_Bisnode_int, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="ROC - Interpreted PD
one year before", add=FALSE)
#Add auc to plot
text(0.5,0.4, "AUC = 0.75", font=2)

#PR AUC
PR_ROC_Bisnode_int  <- performance(predROC_Bisnode_int, "prec", "rec")
plot(PR_ROC_Bisnode_int  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7),  main="PR - Interpreted PD
one year before", add=FALSE)

Bisnode_PR_AUC_int <- as.numeric(performance(predROC_Bisnode_int, "aucpr")@y.values)
Bisnode_PR_AUC_int


#Add auc to plot
text(0.65,0.4, "AUC = 0.53", font=2)


#brier score

brierScore_Bisnode_int <- mean((as.numeric(Predict_bisnode_int)-as.numeric(one_year_be-
fore_bisnode$Default_status_prob))^2)
brierScore_Bisnode_int

#ROC cureve for two years before dataset

Predict_bisnode_int_two <- two_years_before_bisnode[,42] #Column 41 bisnode's rating, 42
is interpreted rating
predROC_Bisnode_int_two<- prediction(as.numeric(Predict_bisnode_int_two), as.nu-
meric(two_years_before_bisnode$Default_status))

Bisnode_AUC_int_two <- as.numeric(performance(predROC_Bisnode_int_two, "auc")@y.values)
Bisnode_AUC_int_two

#ROC AUC
perf_ROC_Bisnode_int_two <- performance(predROC_Bisnode_int_two, "tpr", "fpr")
plot(perf_ROC_Bisnode_int_two, colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="ROC - Interpreted PD
two years before", add=FALSE)
#Add auc to plot
text(0.5,0.4, "AUC = 0.67", font=2)

#PR AUC
PR_ROC_Bisnode_int_two  <- performance(predROC_Bisnode_int_two, "prec", "rec")
plot(PR_ROC_Bisnode_int_two  , colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),text.adj = c(-0.2,1.7), main="PR - Interpreted PD
two years before", add=FALSE)

Bisnode_PR_AUC_int_two <- as.numeric(performance(predROC_Bisnode_int_two,
"aucpr")@y.values)
Bisnode_PR_AUC_int_two

#Add auc to plot
text(0.5,0.2, "AUC = 0.25", font=2)

#brier score

brierScore_Bisnode_int_two <- mean((as.numeric(Predict_bisnode_int_two)-as.nu-
meric(two_years_before_bisnode$Default_status_prob))^2)
brierScore_Bisnode_int_two

#Summary of statistics
Bisnode_AUC #12-month PD ROC AUC
Bisnode_AUC_two #12-month PD ROC AUC two years before
Bisnode_AUC_int #interpreted PD ROC AUC
Bisnode_AUC_int_two #interpreted PD ROC AUC two years before
```

```
brierScore_Bisnode #12-month brier
brierScore_Bisnode_two #12-month brier two years before
brierScore_Bisnode_int #Interpreted brier
brierScore_Bisnode_int_two #Interpreted brier two years before
```