Tomi Himberg

Loan Default Prediction with Machine Learning

Master's Thesis in Information Systems
Supervisor: Dr. Xiaolu Wang
Faculty of Social Sciences, Business and Economics
Åbo Akademi University

Åbo 2021

ABSTRACT

Subject: Information Systems

Writer: Tomi Himberg

Title: Loan Default Prediction with Machine Learning

Supervisor: Dr. Xiaolu Wang

Abstract:

Giving credit is one of the core businesses in banking and the importance of credit risk management was highlighted in the 2008 financial crisis. Increased number of loan defaults was one of the reasons behind the crisis, which led to more regulations in loan granting. Predicting loan defaults has become important as banks try to follow laws and regulations, grant credits to qualified customers, mitigate credits to unqualified customers and to make their application processes efficient. This research studies credit risk in banking, discusses banking regulations which affect loan granting and presents how machine learning is utilized in lending. In addition, the literature review explains machine learning and the steps in building machine learning models. The empirical study is conducted with a loan data set retrieved from Kaggle.com. Predictions are executed with four machine learning algorithms and predictive power is evaluated based on sensitivity, specificity and the area under the ROC curve. The four algorithms used are logistic regression, classification tree, random forest and extreme gradient boosting (XGBoost). Research questions are answered based on the literature review and the results from the empirical study. The results suggest that lenders have various reasons to utilize machine learning in their loan application processes and machine learning enables classifying the majority of qualified and unqualified applicants correctly.

Keywords: Machine learning, Loan default, Loan default prediction, Default risk, Data science, Supervised learning, Predictive analytics, Classification, Logistic regression, Classification tree, Random forest, Extreme gradient boosting, XGBoost

Date: 3.12.2021 | **Number of pages:** 87

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background

Giving credit has a long history as the first documented credits are from ancient Egypt, Babylon and Assyria from 3000 years ago. Credit became more common in the Middle Ages in Europe when merchants travelled throughout the continent trading goods in different countries. The final breakthrough for credit as we know it today came with the Industrial Revolution. The Revolution was quick, and suddenly new products were made and sold to different customers all over the world. Conducting business changed to "buy now – pay later" as businesses could not provide goods and services purely on their profits and needed credit to keep up with orders. In the early 20th century, companies could give credit to a customer whom they knew and trusted. As credit became more common, lenders were forced to know more about their customers and to evaluate their credit amount and length of their loans. Hence, credit management became important. (Bullivant, 2016)

Bullivant (2016) describes credit as the oil of commerce. He elaborates that credit is an essential part of the economy, which enables business growth and personal consumption. Credit helps companies to make investments and expand their businesses, whereas consumers can use credit cards for small purchases or mortgages for buying their own house. However, credit is all about risk, which is unavoidable. Credit risk management is for assessing and managing that calculated risk (Bullivant, 2016). As Bessis (2015) states, risk management has become the core of financial firms as well as insurance companies.

Bandyopadhyay (2016) states that credit risk management is critical for a bank's long-term survival and growth as lending is one of the core businesses in banking. He lists three reasons why credit risk management is important. The first reason is market realities, consisting of non-performing assets, increased competition and collateral values. Non-performing assets are defaulted loans, which increased due to the 2008 financial crisis. Increased competition forces banks to lower their margins, which decreases profits. In addition, volatile collateral values increase credit risk as they affect uncertainty in loan recovery process. The other two reasons are changing regulatory environment and

institution's risk vision. The regulations banks face are involuntary and violating them could lead to immense fines. (Bandyopadhyay, 2016)

Credit risk management is supervised and regulated, which underlines its importance. Bank regulators and financial institutes attempt to mitigate risks to ensure that the financial market functions properly and to avoid losses. Granting credit to every applicant entails major risk in credit defaults or late payments. However, when interest rates are low, banks might be tempted to lower their lending standards to gain more income. As Bandyopadhyay (2016) mentions, credit risk for banks can increase due to changes in business environment. Low interest rates increase competition, which may affect how banks assess their risk-taking capacity. In turn, declining all applicants can have an impact on a bank's reputation and lead to missing potential business opportunities and revenues, as mentioned by Bullivant (2016).

1.1.1 Credit Risk in Banking

The various financial risks banks confront can be broadly classified as credit risk, market risk, liquidity risk and interest rate risk as Bessis (2015) classifies. Bessis (2015) explains credit risk as the risk of a debtor defaulting his or her loan, which leads to losses for the lender. Bandyopadhyay (2016) elaborates that credit risk includes that a group of borrowers or a counterparty fails to meet its obligations, or an investment deteriorates and defaults and explains that loans are the most common source of credit risk for banks. However, financial instruments such as bonds, swaps, options and interbank transactions all include credit risk.

Default risk is a subclass of credit risk, and it is the first extent in estimating credit risk according to Bandyopadhyay (2016). Default risk is the probability that the borrower does not comply with the loan terms and fails to pay back the loan. Banks utilize different methods to estimate default probabilities. The methods include using own data and experience in defaults, mapping internal defaults to external data and using default models. (Bessis, 2015)

1.1.2 Loan Default and Its Impact on Creditors

As stated above, loan default means that a borrower fails to comply with his or her payment obligations to the lender, inducing loss on capital, interest and increase in

collection costs. Loan defaults can be temporary or indefinite. In temporary defaults, the borrower manages to pay back the overdue amount at once or through a payment plan resulting to a partial loss. Default qualifies as indefinite if the payment is 90 days overdue. Losses on indefinite defaults are greater as no interest is paid for a longer time and collection costs increase. (Bessis, 2015).

The Basel II Agreement, which will be introduced in chapter 2, defines loan default on two conditions. First, the lender considers that the borrower is unlikely to pay the loan in full. The Second condition is that the borrower's past due is more than 90 days on any credit. According to the Basel II Agreement, one condition must be met for a loan default. (Bandyopadhyay, 2016) The Basel II Agreement is an important regulation, which ensures that banks are resilient if risks materialize. The agreement obliges financial institutes to calculate credit risk components, which ensure lenders are prepared for potential defaults (Bessis, 2015).

The credit risk components are exposure at default (EAD), loss given default (LGD) and default probability (DP) (Bessis, 2015; Bandyopadhyay, 2016). Default probability is the probability that a borrower will default his or her loan as explained above. Exposure at default is explained by Bandyopadhyay (2016) as the amount of losses the lender may face at a time of default. He elaborates that exposure risk is low with fixed repayment schedules. However, exposure risk with revolving loans is higher as debtors can withdraw or pay credit according to their credit limits.

Loss given default indicates the severity of the loss after a loan default and is the percentage that a bank will not recover after the recovery process (Bessis, 2015; Bandyopadhyay, 2016). Collaterals are used by banks to mitigate losses in case of a default. When a debtor defaults his or her loan, the lender attempts to recover the debt by selling collaterals used in the credit. According to Bessis (2015), the amount recovered is lower than the amount due because of the expenses of the recovery process. In addition, he mentions that when collaterals are realized, the value of the collateral is unknown. Bessis (2015) elaborates that bankruptcy and restructuring loan terms can qualify as defaults as well. He adds that restructuring loan terms is close to default if it is executed due to decrease in the borrower's credit standing, and inability to pay according to current terms.

1.1.3 Field of Study

As introduced above, this study discusses credit risk and its subclass default risk. The field of study is predicting loan defaults with machine learning. Murphy (2012) explains machine learning as a set of methods that detect patterns in data and enables decision making or predictions based on unseen data. In short, machine learning provides automated tools for data analysis. Machine learning is widely applied in lending and predicting default risk as previous research in section 3.1 demonstrate.

1.2 Objective

As mentioned in the previous section, banks face various financial risks, including credit risk. This research focuses on studying default risk, which is one of the credit risk components obliged in the Basel II regulation as explained in section 1.1.2. Since managing credit risk is crucial for banks and calculating default risk is obliged, the objective for this research is to understand how loan granting is regulated, and how machine learning is utilized in loan granting. Regulations in every business field force companies to develop their systems and monitor how they operate. As mentioned in section 1.1.1, credit is all about risk and financial crises have shown that if the risks materialize, the consequences affect the whole economy. Therefore, regulations in lending are tightened and financial institutions must be agile in their operations. Machine learning has become an important tool in following regulations and enabling an agile business environment.

In addition, the objective is to provide a comparison between different machine learning models and to find the most effective model and most important variables in that model. As noted in the previous section 1.1.3, loan default predictions are a widely studied area and previous research have utilized various machine learning algorithms. This research aims to compare more traditional algorithms with more advanced models. The models used in the research are introduced in section 1.3.

To reach the objectives noted above, this study aims to answer four research questions. The four research questions are:

1. How is loan granting regulated?
2. How is machine learning utilized to facilitate decision making in loan granting?

3. Which of the selected machine learning models is the most effective in loan default prediction?
4. Which variables are the most important in predicting a loan default?

1.3 Method

Different methods are used to answer the four research questions. The first research question is answered based on chapter 2, which reviews banking regulations, regulations implemented in Finland and reasons for these laws and regulations. The Second research question is answered based on chapter 3, which consists of explaining how machine learning is utilized in loan granting and how previous research have studied loan defaults. Furthermore, chapter 4 explains different types of machine learning, the model building process and the learning algorithms used in the research. The algorithms used in the study are logistic regression, classification tree, random forest and extreme gradient boosting (XGBoost).

The Empirical study in chapter 5 is conducted to answer the third and fourth research questions. To answer the third question, the learning algorithms mentioned above are compared based on sensitivity and specificity similar to studies by Ince & Aktan (2010) and Lee, Chiu, Chou & Lu (2006). In addition to their research, sensitivity and specificity are maximized with varying the threshold and the AUC is included in comparisons. The fourth research question is answered based on the most powerful learning algorithm. After the most effective model is detected, the most relevant variables of the algorithm are introduced in chapter 6.

As in previous studies, confusion matrix will be used as an evaluation metric. To add to previous studies, threshold will be varied based on Youden's index to maximize sensitivity and specificity and the receiver operator characteristic curve (ROC curve) and the area under the ROC curve will be used in evaluation. Previous studies have used accuracy as an evaluation metric. However, this research presents accuracy but does not include it in evaluation because of the data imbalance.

1.4 The Structure of the Thesis

This study consists of six chapters. The First chapter contains introduction, which offers a brief introduction on the subject, its history and explains risk in lending and loan default, which are important concepts in the research. In addition, the study objective, research questions, method and the structure of the thesis are introduced in the first chapter.

Banking regulations and new regulations in Finland are discussed in chapter 2. In addition, the chapter explains why regulations are important and how they are utilized to secure functional housing markets. Chapter 3 discusses how machine learning is used in loan granting and in the application process. In addition, Chapter 3 introduces various previous research on loan default predictions. The Fourth chapter discusses different machine learning methods, model building steps and learning algorithms used in the empirical study. The chapter explains important concepts such as supervised learning, data pre-processing, performance metrics and learning algorithms logistic regression, classification tree, random forest and XGBoost.

After the important concepts are explained, the fifth chapter explains how they are implemented to the empirical study. Chapter 5 offers an overview of the data set used in the research, how data pre-processing is executed, the model building process and concludes with results and results analysis. To finalize the research project, chapter 6 discusses the results, explains study limitations and gives proposals for future research.

2 BANKING REGULATIONS

Bandyopadhyay (2016) mentions that credit risk management is important due to the changing regulatory environment of banks. The purpose of these regulations is to ensure the financial market is functioning properly and financial crises are avoided. As Bandyopadhyay (2016) notes, the rapid growth of credit in emerging and developed markets lightens lending standards, which can lead to similar financial crisis as in 2008.

"The primary purpose of risk regulations is to prevent systematic risk, or the risk of collapse of the entire system due to interconnections between financial firms." Bessis (2015, p.14). He adds that providing more freedom to financial firms would result in self-regulation and to moral hazard. Bessis (2015) describes moral hazard as a situation where a group is willing to take risks as the result is not borne by that group. Moral hazard decreases trust in financial markets, therefore regulations are needed.

Following the 2008 US mortgage market crisis, the Basel regulators introduced more regulations to make banks more resilient. The Basel II Agreement evolved into the Basel III Agreement. (Bessis, 2015; Magnus, Margerit, Mesnard & Korpas, 2017). The implementation of Basel III has been postponed to 2023, therefore this paper will not discuss it further.

The Basel II Agreement is a capital framework in the European Union. Its purpose is to ensure that banks have enough capital to face risks. Basel II is a structure of three pillars: Pillar 1 sets the rules for minimum capital that banks must possess. The minimum capital is 8% of their risk-weighted assets. Pillar 2 has principles where supervisors can review and ensure that banks have enough capital compared to their risks. Pillar 3 is a market discipline pillar, ensuring that investors get the information to evaluate banks risk profiles. (European Central Bank, 2004). The pillars are presented in figure 1.

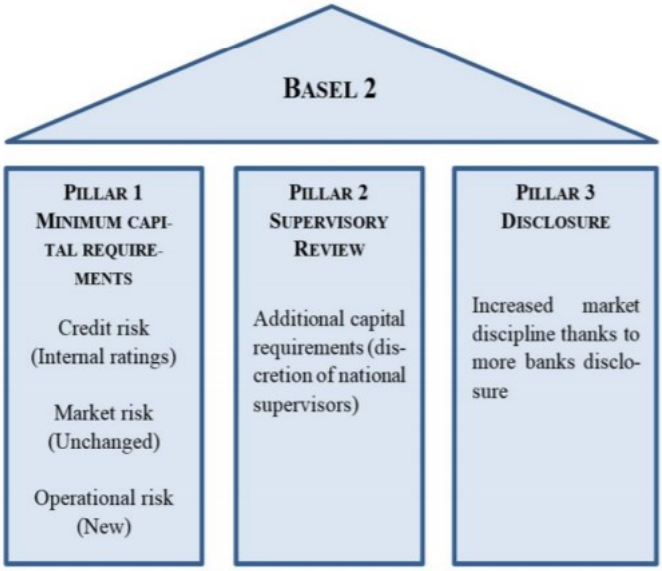

Figure 1. *Basel II pillars*. (The Basel Committee on Banking Supervision (BCBS) as cited in Magnus, Margerit, Mesnard & Korpas, 2017)

The purpose of Basel II is to ensure that financial markets stay stable even during a financial crisis. Every country has its own supervisor, located under pillar 2. (European Central Bank, 2004). In Finland, this supervisor is the Financial Supervisory Authority (FSA) which supervises all banks and lenders and gives guidelines in granting credits. In the past years, the FSA has implemented regulations such as loan-to-value ratio (LTV) and housing loan cap.

2.1 New Regulations in Finland

A major change in the Finnish mortgage granting regulations came in first of July 2016. Then, the housing loan cap and loan-to-value ratio entered into force. LTV refers to the loan amount granted compared to the current value of the collateral at the time of loan granting, whereas loan cap is a macroprudential tool to contain household indebtedness. After the new regulations, banks could grant a maximum loan that is 90% from the current value of the collateral used. For the first time home buyers the admitted LTV is 95%. (Financial Supervisory Authority, 2015.)

To clarify the loan-to-value ratio, we can use a house that costs 100 000 euros as an example: As of first of July 2016, a bank could only grant a loan of 90 000 euros (95 000 euros for a first home buyer) for the same house. The missing 10 000 euros (5 000 euros

for a first home buyer) should be self-financed by the applicant, or he or she must have other collaterals to cover the 10 000- or 5 000-euros loan. The applicant is forbidden to self-finance the missing 10 000 euros or 5 000 euros with another loan. The loan cap set is a maximum as the debtor can self-finance more than 10% or 5%. If the debtor decides to self-finance 50 000 euros of the house in the example, the LTV ratio is 50%.

These regulations affect housing loans, which are granted for buying an apartment or property and loans used for repairing them (Nordea, 2021). Loan-to-value ratio does not affect granting quick loans or consumer credits. The purpose of the loan cap and the loan-to-value ratio is to prevent household indebtedness, stabilize the financing system and secure that the housing market will not overheat. (Financial Supervisory Authority, 2015; Nordea, 2021.)

Since 2016, the loan cap has become a powerful tool for the authorities to adjust the housing markets and indebtedness. The housing loan cap was put to force in 2016 with the ratio of 90% for all but first home buyers. In 2018 the Financial Supervisory Authority was concerned about the household indebtedness and tightened the ratio to 85% but it remained at 95% for first home buyers (Financial Supervisory Authority, 2018). Because of the Covid-19 pandemic, the LTV was again set to 90% to ensure that the housing market keeps functioning. As the Financial Supervisory Authority (2021) stated, the housing market activity rose in the summer of 2021 and therefore the ratio was again tightened to 85% and entered into force on first of October 2021. The adjustments conducted on LTV enhance the challenges banks face and exemplifies how quickly they need to adapt to changes. The changes affect their systems, which need to be modified. In addition, employees and customers must be informed about the adjustments. A customer who negotiated a loan six months ago and received a loan promise, might not be able to buy a house anymore as the loan cap was tightened.

In addition, Finland is changing how lenders need to evaluate borrowers' ability to pay back their loans. Today, lenders check a debtor's credit score and use it in their evaluation. The Ministry of Justice has proposed that Finland would start using a positive credit register in loan application processes. Positive credit register would be a database which is at disposal for all financial institutes, and it would consist of every loan that has been granted to a debtor. Lenders would register every granted loan into the database, and it

would also have information about every payment that is over 45 days past due. (Ministry of Justice, 2021)

2.1.1 Over-indebtedness

One reason to implement the positive credit register introduced above is over-indebtedness. As mentioned above, lenders cannot verify how much credit other financial institutes have granted to the applicant as there is not such database. Banks utilize credit score, bank statements and information from the applicant, which increases the possibility of missing important information about other credits, thus increasing the possibility of over-indebtedness. Over-indebtedness has become an issue and governments have created new laws in order to restrict loan granting and to mitigate risks. One of the reasons behind the US mortgage crisis in 2018 was loan default rate. Due to the raise on federal funds rate, debtors interest rates increased resulting in payment difficulties and defaults. (Amadeo, 2020) In adjustable mortgage rates the interest rates have no cap limit for increase, which results in higher payments. Therefore, over-indebtedness is a risk, which could lead to a vast number of defaults.

Today, numerous lenders offer credit, and it is possible to apply for different loans easily through an online application. The most problematic loan types that Finland has regulated are quick loans. Quick loans are designed to have a short payback time and their interest rates and costs are high. These loans are marketed aggressively and as mentioned by Raijas, Lehtinen & Leskinen (2010), they are targeted toward those who are not able to obtain a loan from their own bank or elsewhere. The same is raised by Statistics Finland (2021). In their study from Finnish household's indebtedness in 2019, they mention that the lower a households' income, the larger the debts compared to the income. In 2019, the Finnish government passed a law, which decreased the maximum interest rates and the cost ceiling that could be charged from quick or unsecured loans (Uusitalo, 2019). In addition, during the Covid-19 pandemic in 2020, the Finnish government passed a temporary law, which restricted the advertisement of quick loans (Finnish Competition and Consumer Authority, 2020). The purpose of these restrictions is to reduce over-indebtedness, especially during the Covid-19 pandemic. Still, the Covid-19 pandemic is not the main reason for these restrictions. Over-indebtedness has increased over the years as credits have become normal in households, consumption has increased and the economy has developed positively (Raijas et al., 2010).

Overall, Finnish households' debts were more than double compared to their annual income. The highest ratio was in single-supporter households. (Statistics Finland, 2021) In 2019, the European Systematic Risk Board (ESRB) gave recommendations to Finland and other countries based on the vulnerabilities on the country's mortgages. For the challenge of limiting household indebtedness, Finland received a recommendation to include a law which limits the loan amount based on the borrower's income (Bank of Finland, 2019).

3 MACHINE LEARNING IN LOAN GRANTING

As a result of the increasing laws and regulations to ensure the functionality of housing markets and to reduce social problems, such as over-indebtedness banks must observe to whom credits are granted. This chapter discusses why detecting risky applicants is important and how machine learning is utilized in lending. Furthermore, the rest of the chapter focuses on previous studies on loan default predictions.

Giving credit is one of the core businesses in banking and banks make profit from interest. In retail banking lending consists of mortgages, credit cards, consumer loans and credits to small enterprises. The banking crises have shown that it is important to distinguish between qualified and unqualified applicants as granting credit to all applicants could result in another global banking crisis. As mentioned by Amadeo (2020), loan default rate was one of the reasons behind the 2008 financial crisis. According to her, deregulation led to debtors taking mortgages that they could not afford and when the federal funds rate was increased, debtors were unable to payback their loans.

In the 2008 financial crisis, 8.8 million jobs were lost in the United States. In addition, unemployment rate spiked to 10%, 8 million homes were foreclosed, and home prices fell 40% on average. In addition, 7.4 trillion dollars were lost in stock wealth. (Silver, 2021) The crisis affected Europe as well: According to Statistics Finland (2013) between 2008-2013, home prices fell in Spain and Greece by 25% and 50% in Ireland. In addition, Finland's unemployment rate in the beginning of 2008 was 6%, which rose to 9% in the early 2009.

Due to the 2008 financial crisis, interest rates have dropped as the European Central Bank has fought against recession (Helenius, 2018). Low interest rates together with digitalization, mean that the number of loan applications banks receive is vast. The increase has led to a point where a bank manager cannot review every loan application manually. In addition, banks must assess every applicant's creditworthiness with his or her credit history or with bank's own experience in defaults. Thus, credit scoring and machine learning have become widely used in banking (Blöchlinger & Leippold, 2006).

AI-assisted automated decisions are used to support decision making and to increase agility and efficiency as Bessis (2015) mentions. However, loan granting is not solely

based on automated decisions and manual labour is still needed. Therefore, as Bessis (2015) notes, making risk processes enterprise-wide is effective and makes loan granting agile. All bank employees should have the same knowledge about loan granting and risk policies, then not every loan application would need to be approved by the highest-ranking credit manager. When the risk policies are clear, the credit manager is able to authorize lower-level managers or advisers in decision making, thus increasing agility and efficiency.

In addition, machine learning facilitates risk mitigation as computers can distinguish qualified loan applications from unqualified. However, a bank cannot become too careful in loan granting and deny all applicants flagged as an unqualified applicant by a machine learning algorithm. Bessis (2015) explains that a careful bank could lose in market shares and revenues as it limits business volume by screening for risky customers. If the risks were not calculated and a bank lent money to everyone, it would have a large market share and revenues. However, detecting risky applicants is important as in the long run a careful bank probably will make more profit than one that does not calculate risks. A careful bank's risks will unlikely materialize and if they do, the risks will have been minimized. Thus, machine learning and manual labour should be utilized together in loan granting. As stated above, detecting risky customers is important and banks are obliged to count default probabilities as mentioned in section 1.1.2. Therefore, loan default predictions are a widely studied area.

3.1 Previous Research

Much research (e.g., Silva, Lopes, Correia & Faria, 2020; Abid, Masmoudi & Zouari-Ghorbel, 2018; Ince & Aktan, 2010) has studied credit risk using different machine learning methods. Some previous research concentrates on only one or two methods (Silva et al., 2020; Abid et al., 2018) and others (e.g., Ince & Aktan, 2010; Lee et.al., 2006) compare various methods and try to find the most effective one.

Silva et al. (2020) studied default risk with logistic regression on a Portuguese credit data set. The data set consisted of 3221 individuals with a 10% default rate. In result analysis, they provided relevant variables and logistic regression's accuracy metric was compared to other studies. Variables "Spread", "Term", "Age", "Credit cards", "Salary" and "Tax echelon" were found relevant and the percentage of correctly classified cases was

89.79%. A sensitivity of 0.94% and a specificity of 99.55% can be counted from the presented confusion matrix.

Abid et al. (2018) studied default risk with logistic regression and discriminant analysis. Tunisian commercial bank's data set consisting of 603 loans was used in the study. The default rate was higher compared to other loan default studies, reaching 56.55%. Their logistic regression model had 99.41% sensitivity and 98.47% specificity. Discriminant analysis provided a sensitivity of 75.36% and a specificity of 59.54%, thus they declared logistic regression to be more powerful. Variables "loan amount", "outstanding loan" and "socio-professional category" were relevant in logistic regression model.

As mentioned above, Ince & Aktan (2010) and Lee et al. (2006) studied default risk with various machine learning models. Both studies created predictions with discriminant analysis, logistic regression, neural networks and decision trees. In addition, Lee et al. (2006) used multivariate adaptive regression splines model (MARS). The data set used by Lee et al. (2006) consisted of 8000 loans from a bank in Taipei, Taiwan. Default ratio of the data was not revealed, but the data set consisted of 9 variables: "Gender", "age", "marriage status", "educational level", "occupation", "job position", "annual income", "residential status" and "credit limits".

To compare the results, Lee et al. (2006) displayed credit scoring results, which consisted of sensitivity, specificity and average correct classification rate. In addition, they compared type I and type II errors. Discriminant analysis had 67.69% sensitivity and 69.91% specificity, logistic regression had a sensitivity of 66.03% and a specificity of 76.38%, whereas neural networks provided a sensitivity of 60.04% and the highest specificity, 89.48%. MARS had highest sensitivity, 69.38%, and its specificity was 86.28%. Finally, decision trees had a sensitivity of 68.29% and a specificity of 87.79%. To conclude their research Lee et al. (2006) stated that the decision tree model and MARS model outperformed other models.

Ince & Aktan (2010) found out that the decision tree model outperformed other models in average classification rate, but neural networks had lower type II errors and therefore had better overall results. Their research used a Turkish bank data set with 1260 loans. The sensitivities and specificities of their study were: Discriminant analysis had a sensitivity of 57.81%, a specificity of 67.09%, logistic regression's sensitivity was

65.06% and specificity 60.10%, neural networks had a sensitivity of 74.10% and a specificity of 51.23%, decision tree's sensitivity was 62.05% and specificity 68.47%.

Chang, Chang, Chu & Tong (2016) compared logistic regression, the Cox model and a decision tree model in their research. They studied loan defaults from a time perspective: The goal was not only to predict which loans will default, but to predict which loans will be defaulted within 12 months after they are granted. In addition, the most important variables in predicting short-term loan defaults were introduced. They mentioned that loans that are defaulted after a short period of time after granting brings great loss to the lender. Evaluation metrics used in their study were sensitivity and precision. Results were compared between the models based on how precisely they predict short-term defaults. Their decision tree model reached a specificity of 81.9% and a precision of 83.3%. Logistic regression had a sensitivity of 46% and a precision of 64.2%, whereas the Cox model achieved 45.5% sensitivity and 45.5% precision on predicting short-term loan defaults. The most important variables in predicting short-term loan defaults were variables "background", "macro", "liability_ratio", "fixed_ratio", "DSR", "competitor_evaluation" and "law".

Previous research have used various metrics for evaluating prediction power, such as classification accuracy, sensitivity and specificity and type I and type II errors. ROC curve was used as an evaluation metric by Silva et al. (2020), but their research did not display the AUC. The area under the ROC curve was presented in loan default predictions by Lessmann, Baesens, Seow & Thomas (2015), Zhu, Qiu, Ergu, Ying & Liu (2019), Xia, Liu & Liu (2017), Wang, Jiang, Ding, Lyu & Liu (2018) and Tian, Xiao, Feng & Wei (2020). However, none of the previous studies used sensitivity, specificity and the AUC together in evaluation.

Lessmann et al. (2015) compared 41 different learning algorithms in their study and utilized eight retail credit scoring data sets. Their study compared individual classifiers (e.g., logistic regression) to more advanced classifiers (e.g., random forest). They mention that logistic regression is an industry standard for credit scoring predictions. In addition, they state that several classifiers, such as random forest performs better than logistic regression. Wang et al. (2018) conducted a default probability study on a peer-to-peer lending data set. They compared ensemble mixture random forest model (EMRF) with a standard mixture cure model, the Cox proportional hazards model and logistic regression

model. They stated that their EMRF model outperformed all the other models based on the mean area under the ROC curve.

Studies by Zhu et al. (2019), Xia et al. (2017), Odegua (2020) and Xu, Lu & Xie (2021) studied loan defaults in peer-to-peer lending. Zhu et al. (2019) created predictions with random forest, decision tree, support vector machine (SVM) and logistic regression. The evaluation metrics used were the ROC curve, the AUC, sensitivity and accuracy. Their study achieved an AUC of 0.983, an accuracy of 98% and a sensitivity of 99% for random forest model. Decision tree had an AUC of 0.958, an accuracy of 95% and a sensitivity of 96%. In addition, SVM had 0.757 AUC, an accuracy of 75% and a sensitivity of 74%. Their logistic regression model had 0.735 AUC, 73% accuracy and 71% sensitivity. However, specificity was not displayed in their research. Xia et al. (2017) compared various algorithms, including logistic regression, random forest and cost-sensitive XGBoost. Their study reached an AUC of 0.5864 for logistic regression, 0.6914 for random forest and 0.7001 for cost-sensitive XGBoost. In addition, Odegua (2020) and Xu et al. (2021) studied loan defaults with XGBoost. They did not display the specificity nor the AUC in their research. Odegua (2020) had a sensitivity of 79% and Xu et al. (2021) had a sensitivity of 96.9%.

Tian et al. (2020) performed their loan default predictions on a data set from a credit assessment company. The data set had 50 000 observations, 350 columns and according to them, more than 70% of the observations were non-defaulted loans. They compared seven machine learning algorithms, including logistic regression, decision tree, random forest and gradient boosting tree. Evaluation metrics used were accuracy, f1 score and the AUC. Their logistic regression model reached an AUC of 0.84, with 74.43% accuracy. Decision tree had an AUC of 0.85, with 84.68% accuracy. Random forest had an AUC of 0.96 and an accuracy of 88.96%, whereas gradient boosting tree reached an AUC of 0.97 and an accuracy of 90.99%.

Yu (2020) studied credit card default prediction with logistic regression, decision tree, adaboosting and random forest. In addition, he created weighted models for each model to overcome data imbalance. The results were compared between each other based on accuracy. Confusion matrices were presented but they were only used to calculate accuracy.

Furthermore, previous research mentioned above did not clarify the threshold used in confusion matrix, which affects sensitivity and specificity. Bessis (2015) explains that banks need to be careful in credit granting in order to limit loan defaults. In addition, they do not want to be too careful as rejecting loans from qualified customers results in lost revenues. Therefore, lenders want to maximize sensitivity and specificity, meaning that they detect as many potential defaults and non-defaults as possible. As Kabacoff (2015) mentions, the threshold used to create the confusion matrix can be varied to maximize sensitivity and specificity.

4 MACHINE LEARNING AND MODEL BUILDING

As mentioned above, competition, the need to improve profitability and detect risky applicants have made loan default predictions a widely studied area and machine learning a crucial part of loan granting. The studies discussed in the previous chapter demonstrate that researchers try to identify the most powerful algorithms in loan default predictions and use different evaluation metrics to assess predictive power. This section introduces different machine learning methods, which are supervised learning, unsupervised learning and reinforcement learning. In addition, section 4.4 explains the steps in building machine learning models and section 4.5 introduces the learning algorithms used in the study.

Everyone creates and consumes data: When you visit a website, buy a product, post to social media or make a phone call, you create data. When scrolling through specialized offers from a local supermarket, one consumes data. In the age of big data, data cannot be processed or analysed manually as before when data was stored in computer centres. The amount of data created daily is vast as everything in human behaviour is captured. (Alpaydin, 2014)

Human behaviour is never completely random and the data for example from grocery shopping leaves a trail. With machine learning, the gathered data is used to predict what customers will buy or might be interested in. Machine learning enables automated methods for data-analysis and makes predictions by learning from previous data. A grocery store uses the prediction for specialized offers and ditto YouTube offers recommended videos based on the videos watched earlier. These offers and recommendations are created with machine learning from massive data sets. (Alpaydin, 2014)

Machine learning is divided into three categories: Supervised learning, unsupervised learning and reinforcement learning (Alpaydin, 2014; Murphy, 2012). The categories and their differences will be introduced next.

4.1 Supervised Learning

In supervised learning, the learning of computers and algorithms occurs with past experiences or using example data (Alpaydin, 2014). Yang, Chen, Zhang, Park & Yoon (2018) explain that the example data (or the training data) is used in teaching computers based on the input vectors and the corresponding dependent variables. After the learning process is finished, the model is used with unseen data and evaluated.

Supervised or predictive learning has two types of problems: Classification or regression problems. Murphy (2012) explains that in classification problems the output or the response variable is a categorical or nominal value and in regression problems the response variable is numerical. In addition, Murphy (2012) mentions that the input and output variables can be in the form or an image, a sentence or a graph, but most methods assume them to be categorical or numerical. For example, predicting a loan default where the response variable is "good customer", or "bad customer" is a classification problem. Predicting a person's income level or a car's price exemplifies a regression problem.

Alpaydin (2014) and Murphy (2012) explain that the aim in supervised learning is to learn a mapping from input to an output, which has correct values from a supervisor. For loan default prediction, the input consists of the information about the customer and the loan, and the classifier classifies the input to "good customers" or "bad customers". Classification can also be used with probabilities. In some cases, it might be of interest to count the probability of a customer defaulting a loan. Then, if the customer has for example 60% probability to default the loan, the creditor decides whether to grant or reject the application. (Alpaydin, 2014)

4.2 Unsupervised Learning

Unsupervised learning differs from supervised learning as it has only input data. Alpaydin (2014) explains that supervised learning uses output data to find a mapping from input data, whereas unsupervised learning does not have output data and the aim is to find regularities from the input. Murphy (2012) elaborates that the unsupervised learning problem is not as well-defined and is also known as descriptive learning or knowledge discovery. There is not an error metric for unsupervised learning as there is not output data to show what kind of results to search.

Alpaydin (2014) offers an example on unsupervised learning: A company is interested in expanding their business and they have data about past customers, their demographic information and transactions. With a clustering model they are able to assign similar customers into same groups. When similar customers are in same groups, they can detect outliers which in this example are the customers who are different from the others and do not fit any group. After detecting outliers, they are able to concentrate their marketing campaign to these people and expand their business.

4.3 Reinforcement Learning

Murphy (2012) mentions that reinforcement learning is a third type of machine learning, but it is not as commonly used as supervised or unsupervised learning. Reinforcement learning is a way to learn with rewards or punishments. Alpaydin (2014) and Yang et al. (2018) elaborate that in reinforcement training, the training occurs with actions (reward or punishment) and not with output data. As exemplified by Alpaydin (2014), if a robot has an unfamiliar environment and it has to find a goal in that environment, reinforcement learning is used. The robot is able to go anywhere in its surroundings, but it is given rewards or punishments as a feedback on its decisions. Based on the feedback the robot learns and it will eventually find the fastest route to the goal.

All machine learning types mentioned above use machine learning algorithms in learning. The route from a raw data set to having a prediction model or a robot learning with reinforcement learning is complex. Data must be pre-processed and machine learning models must be built, and the phases are introduced next.

4.4 Model Building

As mentioned in section 4.1 predicting a loan default is a predictive learning problem. In predictive analytics, the aim is to extract information from large data sets and predict possible future outcomes as mentioned by Larose & Larose (2015). Shmueli & Koppius (2011) elaborate that in addition to making predictions, predictive analytics consists of methods for evaluating predictive power. According to them, predictive power refers to a model's ability to create predictions on new observations.

Figure 2 exemplifies the building of predictive or explanatory models. Shmueli & Koppius (2011) mention that while the main steps of creating predictive and explanatory models are the same, predictive models use different criteria in each step. The empirical study of this research follows the steps presented in figure 2.

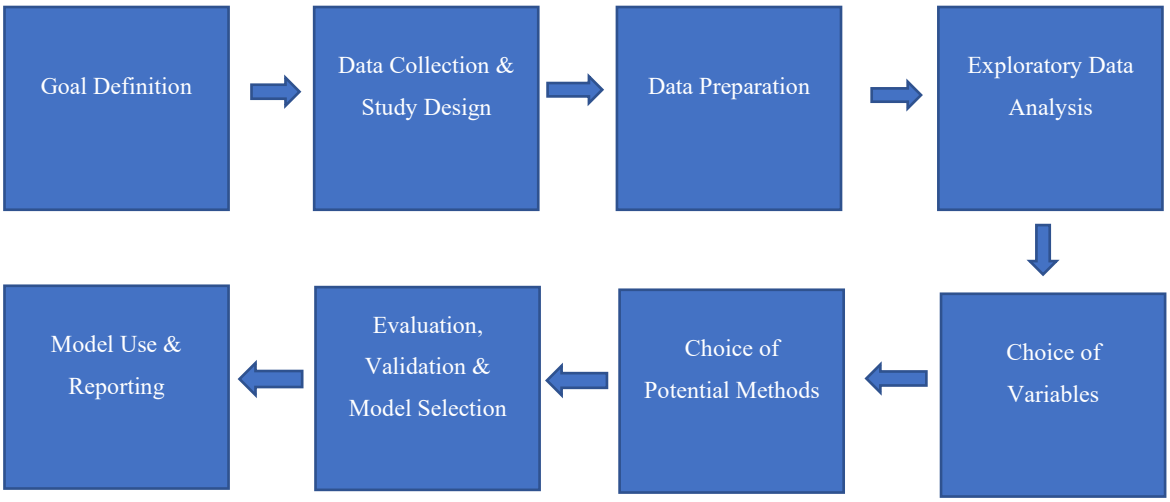

Figure 2. *Model building steps*. (Shmueli & Koppius, 2011, p.563)

4.4.1 Data Pre-processing

Data preparation is a crucial phase in machine learning. Larose & Larose (2015) mention that the raw data is usually incomplete, unprocessed and contains obsolete data, missing values, outliers and data in wrong form for the model. The aim of pre-processing data is to minimize garbage going into the model and to minimize the garbage coming out of the model (Larose & Larose, 2015; Mueller & Massaron, 2016). Mueller and Massaron (2016) compare data pre-processing to building a foundation in house building. The house can have a beautiful architecture but if the foundation is rotten, the house has no value. In addition, they mention that around 80% of the time on a machine learning project is spent on data cleaning.

4.4.1.1 Missing Values

Missing values correction is the most time consuming and problematic step in data cleaning. It is also highly important to handle missing values as they make it difficult for the algorithm to learn in training. Missing values are values typically coded as NA, null or as an empty cell, but missing values can have any type of value, e.g. the value of 0. (Abbott, 2014; Mueller & Massaron, 2016)

A common method of handling missing values is deleting the values. The challenge is that in machine learning, usually more information is better (Larose & Larose 2015). Deleting missing values might lead to deleting important information that has importance in the model as Abbott (2014) mentions. If removing missing values leads to bias in the data set or one aims to keep all the information in the data set, imputing missing values is an option. Missing value imputation means changing missing values to a possible value that represents the variable. Imputation can be executed with a constant, e.g., mean or median, imputing with a value outside the normal value range, imputing with the number 0, distributions or by imputing with values from other features (Abbott, 2014; Mueller & Massaron, 2016).

It is important to understand the data when handling missing values. As mentioned above, deleting all missing values might result in deleting important information or too many values for the model to work. In addition, Mueller & Massaron (2016) note that even if handling missing values with a mean is easy and common, it is not optimal in some cases. They offer an example on studying income levels in a population. Wealthy people tend to hide their true income level because of privacy, while poor people might not want to reveal their income level for fear of negative judgment. If one uses mean to replace missing values in such case, the values are not representative to the population.

4.4.1.2 Outliers

Outliers or anomalies are data points that differ from the expected values, and one is certain they are not correct. An outlier may indicate unusual behaviour. For example, an outlier in credit card transactions could indicate fraud, or it could be an error made in saving data. Outliers are problematic in machine learning as machines learn from examples of data. Therefore, outliers might undervalue the normal values and have a vast impact e.g., on the mean. (Alpaydin, 2014; Mueller & Massaron, 2016)

Outliers can be detected with exploratory data analysis (EDA). For example, an immense difference between the mean and the median might indicate outliers in the data. A numerical method is to use the z-score standardization as mentioned by Larose & Larose (2015) and Mueller & Massaron (2016). In z-score standardization, a data point is an outlier if the z-score is for instance -3 or 3 standard deviations from the mean. Another effective way of detecting outliers is visualization. For example, histograms, boxplots and scatter plots give an understanding on outlier values (Larose & Larose, 2015; Mueller &

Massaron, 2016). Figure 3 exemplifies a scatter plot with two outliers. In the EDA for this research, outlier detection was executed with the z-score standardization and with visualisation, therefore these methods are introduced. Outlier handling in this study is explained in section 5.3.4.

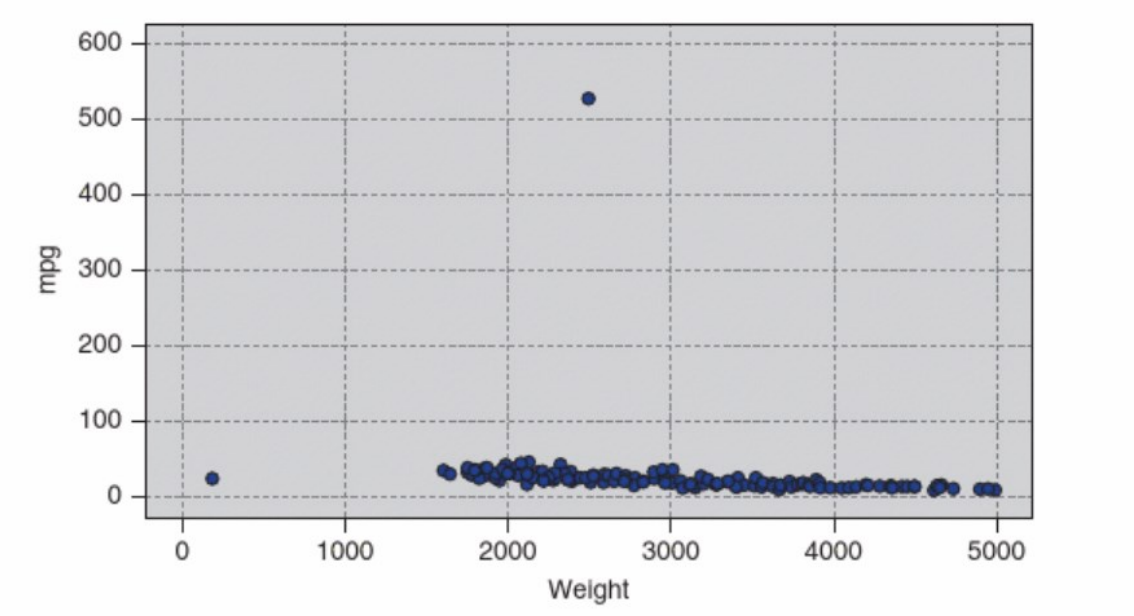

Figure 3. *Scatter plot indicating two outliers*. (Larose & Larose, 2015, p.27)

4.4.1.3 Reclassifying Categorical Values

Some machine learning algorithms, such as logistic regression and decision trees, do not work optimally if a predictor variable has too many unique values. In numerical variables, for example age usually has various values. To optimize the algorithm, numeric variables can be partitioned into bins. For example, variable age can be divided into bins of 20-29 years old, 30-39 years old, 40-49 years olds etc. (Larose & Larose, 2015)

Larose & Larose (2015) elaborate that the problem is common in categorical values and reclassifying these values can be executed similarly. They offer predictor "state", which refers to the number of states in the USA as an example. The predictor would have 50 unique values. "State" could be reclassified to "Northeast", "Southeast", "North Central", "Southwest" and "South". After reclassifying, the algorithm has to handle five unique values instead of 50, which optimizes the algorithm's performance.

4.4.1.4 Feature Creation

The algorithm's performance can also be improved with new features. The data available might have variables which have no predictive power separately but are powerful when added together. The algorithm cannot link these variables together and learn, therefore a new variable must be created. For example, when modelling the price of real estate properties, the size of the property (square meter) usually affects the price. If the information on the size of the property is given in terms of two separate variables, consisting of each side's length, the algorithm cannot calculate these lengths together. (Mueller & Massaron, 2016)

Feature creation can be used to overcome this problem. To calculate the size of the property, the side lengths are multiplied together and put to a new variable, for example "Square meter" to gain predictive power. Mueller & Massaron (2016) mention that the method to create a new variable from existing data is called deriving a new variable. They add that knowing the problem well facilitates in feature creation as one understands how a human would solve the problem. In addition, common knowledge or expertise in the field helps in feature creation and therefore enables improving the model's predictive power.

4.4.1.5 Correlated Variables

Correlation is explained by Larose & Larose (2015) with two variables, X and Y. The two variables are correlated if an increase in X results in increase in Y. To exemplify, a car's engine power is correlated to its fuel consumption. However, more than two variables can have high correlation resulting in multicollinearity (Mueller & Massaron, 2016). Larose & Larose (2015) state that correlated variables should not be used in statistical models as they might overemphasize one data point or make the model unreliable. Mueller & Massaron (2016) elaborate that an ideal machine learning model has variables which do not correlate completely. However, they remind that in reality variables often correlate to each other.

Correlation is stated as a value called correlation coefficient. Correlation coefficient is stated with a letter r ranging from -1 (negative correlation) to +1 (positive correlation) as explained by Brown (2014) and Larose & Larose (2015). Brown (2014) elaborates that r of -1 or +1 means that the variables have perfect correlation, whereas r of 0 implies

uncorrelation. Nickolas (2021) explains that if variables have perfect positive correlation, an increase or decrease in one variable results in an increase or decrease in the other variable with the same magnitude. He adds that perfect negative correlation results in variables moving to opposite directions: An increase in one variable results in decrease in the other and vice versa. As high correlation could lead to an unreliable model, correlated variables should be handled. Larose & Larose (2015) mention that if two variables have perfect correlation, one should be omitted from the data. In addition, multicollinearity should be identified and dimension-reduction methods should be used.

4.4.1.6 Normalization

Numeric variables tend to have wide range in their values. In a credit data set, for example variable age has a smaller range compared to the income variable. Age could range from 18 years old to 90 years old, while yearly income could have a range of 0 to a million. Without normalization, the yearly income would have a greater impact on the results than age, as yearly income has wider variability. With normalization this bias is removed. (Larose & Larose, 2015)

The most commonly used normalization methods are min-max normalization and z-score standardization. Mueller & Massaron (2016) explain that min-max normalization scales the values from 0 to 1, by removing the minimum value and dividing it by range (maximum value minus minimum value). Abbott (2014) elaborates that the min-max normalization presents the percentage of the value relative to its maximum. In z-score normalization, most of the values will be within the range of -3 and 3, while the mean is centred to 0. Scaling works by taking the difference between the variable value and the column mean value, which is divided by the standard deviation of the column values. (Larose & Larose, 2015; Mueller & Massaron, 2016)

4.4.1.7 Imbalanced Data

Data imbalance is common in many real-life situations. Loan defaults and if a patient has a cancer or not, are examples of real-life situations with imbalanced classes. Usually, the important class (minority class) has fewer values than the other class (majority class). Data sets are most often imbalanced, but the severity of the imbalance differs. (Brownlee, 2019)

As most machine learning algorithms are designed to learn from data sets that are equally divided, severe imbalance impacts the algorithm's ability to learn. The algorithms learn from examples and if the classes are severely biased, the majority class impacts more than the minority class. (Brownlee, 2019.) Oversampling and undersampling are techniques to overcome data imbalance and they are explained in section 4.4.1.7.

Data imbalance must be noted when evaluating model performance. As Awad & Khanna (2015) mention, accuracy is a biased metric if the data set is imbalanced. Galar, Fernández, Barrenechea, Bustince & Herrera (2012) elaborate that accuracy has been the most used metric in evaluating model performance, but it is not a valid metric for imbalanced data sets. As explained by Galar et al. (2012) a data set might have an imbalance ratio of 1:100. Meaning, that for one example of minority class there are 100 examples of majority class. In such case the model could have an accuracy of 99% if it would classify all of them into the majority class. For imbalanced data sets the ROC-curve and the confusion matrix with true positive and true negative rates are better evaluation metrics than accuracy (Awad & Khanna, 2015). The ROC-curve and the confusion matrix are introduced in the section 4.4.3.

4.4.1.8 Oversampling and Undersampling

The problem of imbalanced data can be handled with oversampling and undersampling the data. In oversampling, the minority class gets a higher weight to balance the imbalance ratio. When a data set is undersampled, the amount of majority class is randomly decreased to balance the ratio. (Tian et al., 2020.) Weighing is executed by adding and removing objects. In oversampling objects are added to the minority class and in undersampling objects are deleted from the majority class, as elaborated by Krawczyk (2016). He adds that a challenge in oversampling is that it might add insignificant objects to training data, whereas undersampling might delete significant samples, which effect predictions. Another challenge noted by Pykes (2020) is the computational cost, particularly in oversampling. As oversampling increases the number of objects in the majority class, executing the algorithm takes more time and computer memory.

Brownlee (2021) mentions that oversampling and undersampling can be applied to binary classification problems and multi-class classification problems. In addition, as these methods do not assume anything about the data, they are referred to as "naive

resampling". He adds that oversampling and undersampling are only applied to training set and test set is used only for evaluation.

4.4.2 Model Validation

After the data is pre-processed, the model must be validated. Model validation is an important step in machine learning as without it, one has no information of how the model works on unseen data. If data is properly validated, the model can be tested right after model training and if it is not performing, training continues (Ramzai, 2020). Two model validation techniques, the validation set approach and the k-fold cross-validation, are introduced in this section. Both methods were used in the empirical study.

4.4.2.1 The Validation Set Approach

The validation set approach is a method to split the data into training and test set and in some cases to a third, validation set. The split is executed to ensure that the model is working properly on unseen data after the model is built. In real-life machine learning problems, if a whole data set is used in training the model, fresh data is rarely available for testing. (Mueller & Massaron, 2016)

The data split is performed to have the training set to train the model and the test set for evaluation. The test set is used as the unseen data and is put aside until evaluation. The split should be performed randomly to prevent underestimation and overestimation, particularly with imbalanced data sets. Usually, 25 to 20 percent of the observations are assigned to testing and 75 to 80 percent to training. The test set can also be used to tune the algorithm and if it is used, the split usually is 70 percent to training, 20 percent to testing and 10 percent to validation set. (Mueller & Massaron, 2016)

As Abbott (2014) points out, to use the validation set approach, the data set must have enough observations for it to be split and the target variable must be well distributed. Both of these problems lead to statistical insignificance, which leads to underperforming. If the data set has these problems, other resampling methods such as the k-fold cross-validation should be used.

4.4.2.2 K-fold Cross-validation

Mueller & Massaron (2016) mention that if the data set has over 10 000 observations, the validation set approach can be used confidently. For a smaller data set they prefer considering other resampling methods. In smaller data sets, the validation set approach could leave out some useful information because of the split. For imbalanced data sets, the challenge is that the training set could have a majority of the positive values and only a few in test set, which would bias the results (Larose & Larose, 2015). Abbott (2014) elaborates that the advantage in k-fold cross-validation is that it uses all the data in model building. As all the data is used, all information is available equally for the training and evaluation.

The k in k-fold cross-validation refers to the number of subsets or folds used in training and testing the model. The k can be any number but using 5-fold or 10-fold is a popular choice. First, the data is randomly split into k-folds of approximately equal size. Next, every subset is assigned a role, the first fold is used as the test set, while the other k-1 folds are used as training. These steps are executed for k times until every fold has been used as the test set once. The result is k different models, and the result is combined by averaging the test error. (Abbott, 2014; Larose & Larose (2015). The validation method is visualised in figure 4.

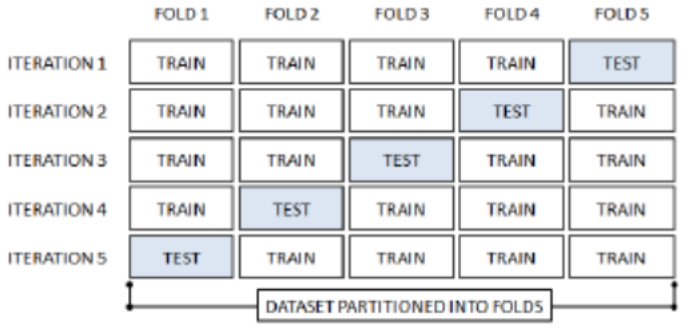

Figure 4. *Visual presentation of the k-fold cross-validation*. (Mueller & Massaron, 2016, p. 192)

Mueller & Massaron (2016) offer a summary on the benefits of k-fold cross-validation: Cross-validation works despite the number of examples in the data set as by varying the number of folds, the size of training set and test set can be varied. The difference in distribution does not matter as much as the method uses every fold as a test set and a training set. Furthermore, with k-fold cross-validation all the observations are tested and

the whole data set is used. In addition, Mueller & Massaron (2016) mention that the predictive performance can be estimated by taking the mean from the cross-validation results.

4.4.3 Performance Metrics

The data used in the research is a loan data set, which is heavily imbalanced. As mentioned in the section 4.4.1.7, accuracy is not a proper metric for imbalanced data sets. This section explains model evaluation metrics, the ROC curve and the confusion matrix that are suitable for imbalanced data sets and were used in the research. To understand how the ROC curve and the confusion matrix operates, also sensitivity and specificity are introduced.

4.4.3.1 Confusion Matrix

In a binary classification problem as a loan default, the model can make an error in two ways: It can predict a default when the actual value is not a default and predict a non-default when the value is default. The confusion matrix is a 2x2 matrix that offers detailed information about classification errors by revealing the two types of errors and the two types of correct predictions, totalling in four outcomes. (Abbott, 2014)

The four outcomes are true positive (TP), false positive (FP), false negative (FN) and true negative (TN) which are displayed in figure 5. If a binary problem has two possible values, Y or N, TP shows that the predicted class is Y, and the actual value is Y. FP displays that the predicted class is Y and the actual class is N. In FN, the predicted value is N, and the actual value is Y. TN indicates that the predicted class is N, and the actual value is N (Kotu & Deshpande, 2014). Several metrics, such as sensitivity and specificity, can be calculated with these values.

Table 8.1 Confusion Matrix

		Actual Class(Observation)	
		Y	**N**
Predicted Class (Expectation)	Y	**TP** (true positive) Correct result	**FP** (false positive) Unexpected result
	N	**FN** (false negative) Missing result	**TN** (true negative) Correct absence of result

Figure 5. *Confusion matrix*. (Kotu & Deshpande, 2014, p.259)

4.4.3.2 Sensitivity and Specificity

The ROC curve and the confusion matrix use sensitivity and specificity in evaluating the model. Kabacoff (2015) explains that sensitivity is the probability of obtaining a positive classification when the actual value is positive, and specificity is the probability of obtaining a negative classification when the actual value is negative. He adds that sensitivity is also known as the true positive rate or recall and specificity as true negative rate. To clarify, in loan default prediction sensitivity indicates how well the model identifies customers with loan defaults and specificity displays how the model identifies customers who have not defaulted. Kabacoff (2015) elaborates that sensitivity is calculated by dividing true positives with the sum of true positives and false negatives as presented in eq. (1). Specificity is calculated by dividing true negatives with the sum of true negatives and false positives as in eq. (2). The formulas below are cited from Larose & Larose (2015, p.457).

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{Number\ of\ true\ positives}}{\boldsymbol{Total\ actually\ positive}} = \frac{\boldsymbol{TP}}{\boldsymbol{TAP}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (1)$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{Number\ of\ true\ negatives}}{\boldsymbol{Total\ actually\ negative}} = \frac{\boldsymbol{TN}}{\boldsymbol{TAN}} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}} \quad (2)$$

4.4.3.3 Youden's Index

The values on sensitivity and specificity can be varied to improve the classification model by changing the threshold or cut-off value. For example, if threshold value is 0.5, a customer with a default probability of below 0.5 is assigned to negatives and above 0.5 to positives. By varying the threshold value, sensitivity can be increased or decreased with the expense of specificity and vice versa. (Kabacoff, 2015)

Yin & Tian (2014) mention that there are several cut-off point estimation methods, but the Youden's index is the most used. Youden (1950) as cited by Yin & Tian (2014), states that the Youden's index finds the cut-off point that maximizes sensitivity and specificity. The Youden's index expanded formula below in eq. (3) is cited from Youden (1950, p.34). In the formula, first sensitivity and specificity are summed together and 1 is deducted from their sum. The number of true positives is a, b stands for false negatives, c is false positives and d is true negatives.

$$J = \frac{a}{a+b} + \frac{d}{c+d} - 1 \quad (3)$$

Lenders do not want to grant loans to applicants who are likely to default nor want to decline loans from applicants who will not default. Hence, their objective is to maximize sensitivity and specificity. Based on this, the Youden's index is employed to determine the cut-off value in the research.

4.4.3.4 ROC Curve

In addition to confusion matrix, the receiver operator characteristic curve (ROC curve) is another tool to assess the classification models quality. Assessing a model's quality can be executed with only one metric, e.g., accuracy, if it is clear what is the best evaluation metric for the situation. Often times, it is difficult to specify one metric. The ROC curve is commonly used as it summarizes performance over all possible confusion matrices. As the range of conditions is wide, ROC curve is applicable also for imbalanced data sets. (Krzanowski & Hand, 2009; Kotu & Deshpande, 2014)

The ROC curve is created by plotting true positive rate on the y-axis and false positive rate on the x-axis. The formula for counting false positive rate is presented in eq. (4) and false negative rate is presented in eq. (5). The formulas are cited from Larose & Larose (2015, p.458). As mentioned above, the ROC curve is a visualization of all possible confusion matrices, where all true positive rate and false positive rate pairs are found with threshold varying from 0 to 1. (Abbott, 2014). Krzanowski & Hand (2009) elaborate that because the threshold varies, the ROC curve can be seen as a complete representation of classifier performance.

$$\textbf{False negative rate} = 1 - \textbf{sensitivity} = \frac{FN}{TAP} = \frac{FN}{TP+FN} \quad (4)$$

$$\textbf{False positive rate} = 1 - \textbf{specificity} = \frac{FP}{TAN} = \frac{FP}{FP+TN} \quad (5)$$

A perfect classifier would have a true positive rate of 1 and a false positive rate of 0. In such situation the ROC curve is at its highest, nearing the top left corner where the true positive rate is 1. The worst result would be when true positive rate is 0 and false positive rate is 1, which locates in the bottom right corner. On the diagonal line the true positive rate and the false positive rate would be equal, as mentioned by Alpaydin (2014). Figure

6 exemplifies a ROC curve, an ideal ROC curve and a random ROC curve meaning that the false positive rate and the true positive rate would be equal. The ROC curve takes every sample on the curve, thus the more samples it has, the smoother the curve will be. In figure 6, the number of samples is low as the steps are visible and in figure 7, the number of samples is high as the curve is smoother.

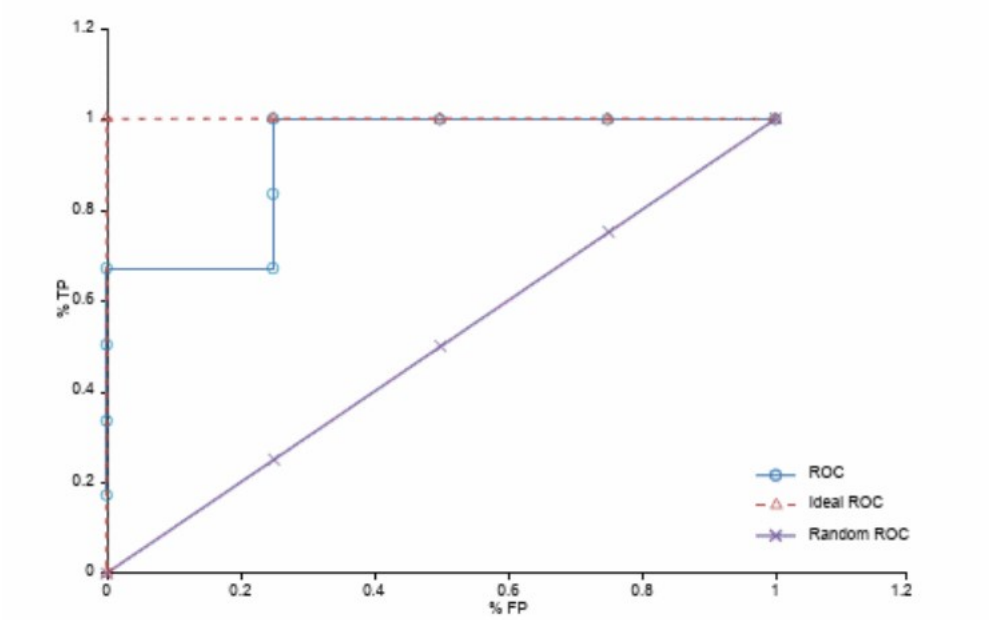

Figure 6. *The ROC curve*. (Kotu & Deshpande, 2014, p.262)

The ROC curve can be complemented with a single-numeric metric called the area under the ROC curve (AUC). It is one of the most common metrics in comparing classification models, as stated by Abbott (2014). He elaborates that the AUC is the area between the coordinates (0,0) and (1,1) under the ROC curve. The perfect model would have an AUC of 1, and the curve would stretch towards the upper left corner. The ideal ROC curve in figure 6 has an AUC of 1. A random classifier would have an AUC of 0.5 and the ROC curve would locate on the diagonal line. The random ROC curve in figure 6 has an AUC of 0.5. To clarify, figure 7 displays the AUC. The red curve is the ROC curve and the blue area between coordinates (0,0) and (1,1) under the ROC curve is the AUC.

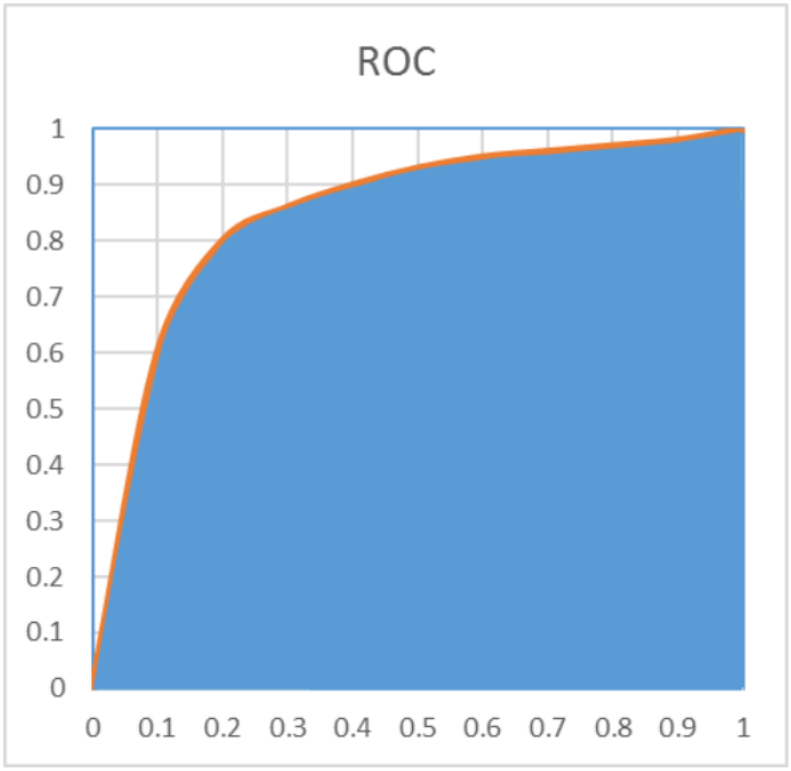

Figure 7. *The area under the curve (AUC).* (Gonzalez, 2017. Retrieved from Medium)

4.4.4 Hyperparameters

Hyperparameter tuning is used to increase a machine learning model's predictive power. Mueller & Massaron (2016) compare hyperparameter tuning to a parent helping a child to draw a tree. A child tries to draw a tree, with his or her own experience and knowledge but needs help from parents, who offer more knowledge and guides in drawing a tree. Mithrakumar (2019) elaborates that hyperparameter tuning is challenging as there is no direct way to calculate how tuning affects the model. He explains that the process starts with testing possible values for all hyperparameters and tuning them based on trial-and-error process. Mithrakumar (2019) mentions that after the range of values is decided hyperparameter tuning methods, such as Grid search or Bayesian Optimization can be appointed to the model.

Hyperparameter tuning is used for more complex algorithms (Mueller & Massaron, 2016). Logistic regression does not require hyperparameter tuning. However, hyperparameter tuning is applicable to tree-based algorithms, such as decision trees, random forest and XGBoost. Overfitting and underfitting are challenges with tree-based algorithms. Too large tree can result in overfitting and a small tree is prone to underfitting,

which decrease predictive power. Hyperparameter tuning can be utilized to overcome that problem. As Mithrakumar (2019) mentions tree depth and the minimum number of samples to split an internal note are commonly used hyperparameters to tune in decision trees.

For random forest model, hyperparameters such as “ntree” and “mtry” can be tuned. “Ntree” stands for number of trees in the forest and “mtry” indicates the number of variables randomly sampled as each split’s candidates, as stated by Brownlee (2016b). He elaborates that random forest has many hyperparameters that can be tuned but “ntree” and “mtry” have major impact on results. XGBoost has similar hyperparameters compared to decision trees and random forest. Hackerearth (n.d.) explains that XGBoost’s “nrounds” is similar to number of trees to grow and tree depth can be tuned in XGBoost as well. In addition, hyperparameter “gamma” controls overfitting in XGBoost and “eta” controls the model’s learning rate.

As mentioned above, hyperparameter tuning can improve machine learning model’s predictive power. However, as Mueller & Massaron (2016) mention too much tuning can decrease the performance as the algorithm starts to detect false signals from the data set. Therefore, as learning algorithms have various hyperparameters, it is important to understand how they affect the learning process.

4.5 Learning Algorithms

Learning algorithms in machine learning are the problem solvers. The algorithms affect results based on the inputs and their goal is to solve the problem based on what they have learned. As Mueller & Massaron (2016) mention, the algorithms modify how the computer interprets the data. The purpose of this section is to explain the algorithms used in the research. The four learning algorithms are logistic regression, classification tree, random forest and XGBoost.

4.5.1 Logistic Regression

As mentioned in section 4.1 supervised or predictive learning has two types of problems: Classification or regression problems. The aim in supervised learning is to learn a mapping from an input to an output, which has correct values from a supervisor

(Alpaydin, 2014; Murphy, 2012). Hosmer, Lemeshow & Sturdivant (2013) mention that logistic regression is the most often used method in describing the relationship between the dependent (outcome or response) variable and one or more independent (predictor or explanatory) variables. As mentioned in section 3.1, studies such as Silva et al. (2020), Yu (2020), Abid et al. (2018), Lee et al. (2006), Ince & Aktan (2010), Zhu et al. (2019), Xia et al. (2017) and Tian et al. (2020) all applied logistic regression in their loan default studies.

As Hosmer et al. (2013) state, the logistic regression model differs from linear regression as in logistic regression the outcome variable is binary. Therefore, logistic regression is used as a classification method, not as a regression method as the name suggests. In binary problems, the outcome variable can have two values: True or false that are usually coded as 1 and 0. In a data set, the outcome variable could be the ability to pay back a loan or default, win or lose or sick or healthy (Abbott, 2014). A logistic regression equation is presented in eq. (6) and cited from Anderson, Sweeney & Williams (2010, p.684). The value of E(y) is the probability that y = 1 given a particular set of values for the independent variables (Anderson et al. 2010).

$$\boldsymbol{E(y) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_\rho x_\rho}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_\rho x_\rho}}} \quad (6)$$

In logistic regression, the probabilities are displayed by a S-shaped logistic curve (figure 8). The logistic curve is bounded by the values 0 to 1 which makes the edges of the graph scale to the bounded values. As in figure 8, if the input value is 2 to 3, the growth is almost linear. However, the logistic regression equation is nonlinear. When the input value is 3 to 4 the probability growth decreases. Figure 8 corresponds to a logistic regression equation, when there is only on independent variable. As Anderson et al. (2010) mention, if the model has two independent variables, the curve is a 3-dimensional multiple regression equation.

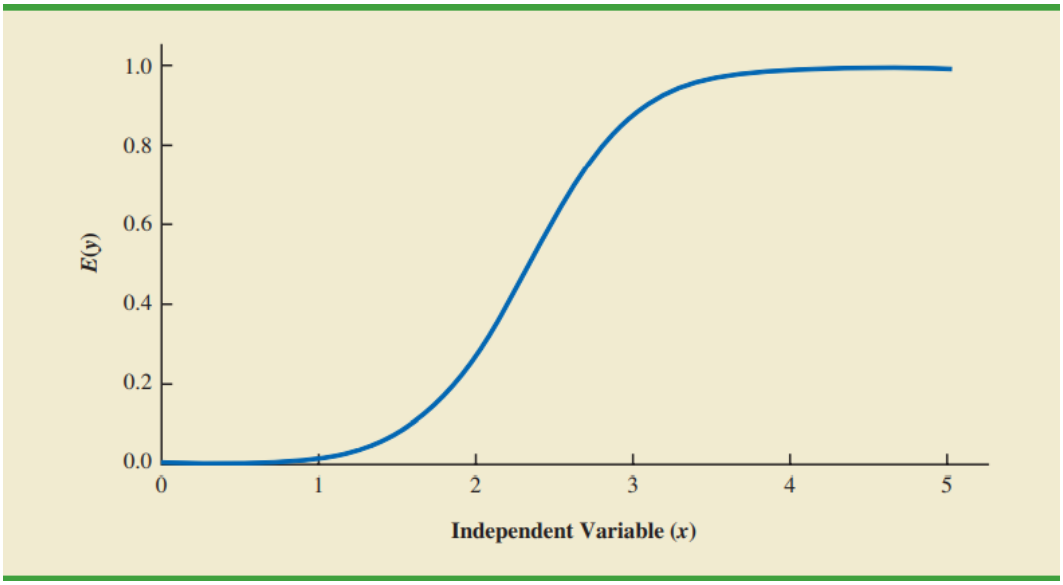

Figure 8. *The logistic curve*. (Anderson et al. 2010, p.685)

4.5.2 Classification Tree

Decision trees are used to predict binary outcome variables with predictor variables. Predictor variables are used to create binary splits, which create a tree that is employed to classify new observations into two groups as explained by Kobacoff (2015). Decision trees can be used in regression and classification analysis. James, Witten, Hastie & Tibshirani (2013) explain that regression trees and classification trees are similar, but the classification tree is employed to predict a qualitative response and regression tree a quantitative one. They elaborate that the prediction in classification tree is executed based on the most commonly occurring class. In regression trees the predicted class is given as the mean of observations that belong to the same terminal node.

Every tree model is constructed from nodes and each node represents one of the input variables. The base of a tree is a root node, and the value of an input variable runs down from the root node to a decision node and ends up in terminal leaf. At every decision node, a decision is made based on impurity measures on which branch to take. (Alpaydin, 2014)

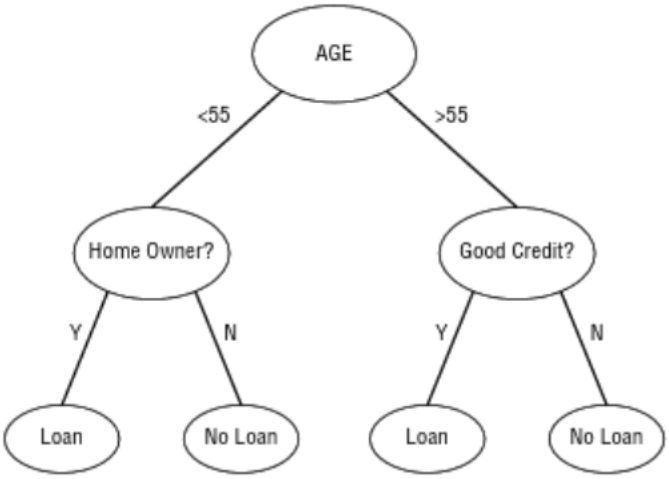

Figure 9. *Classification tree*. (Bell, 2014, p.48)

Figure 9 displays a decision tree. In the tree, "Age" is the root node, "Home owner?" and "Good credit?" are the decision nodes. At the bottom are the leaves containing the outcome value. First the tree checks if the person applying for a loan is over or under 55 years old. If he or she is under 55 years old, the tree checks if the applicant is a home owner or not. If not, loan is not granted and if yes, loan is granted. The same decision is made for over 55 years old's, based on good or bad credit.

Bell (2014) mentions that decision trees are used within many industries such as financial institutions and marketers. He states that decision trees are popular as they are easy to read and easy to explain to others. In addition, decision trees handle numerical and categorized values, and big data sets well. Due to these advantages, decision trees are commonly used in studying loan defaults.

Lee et al. (2006), Ince & Aktan (2010), Yu (2020), Brown & Mues (2012) and Chang et al. (2016) utilized decision trees in their loan default studies. Lee et al. (2006) and Ince & Aktan (2010) applied CART algorithm as they mentioned it is the most commonly used algorithm for decision trees. Brown & Mues (2012) and Chang et al. (2016) applied a different decision tree algorithm, C4.5 classifier in their studies. Brown & Mues (2012) explain that the C4.5 examines information gain that results from an attribute used to split the data and the attribute with the highest information gain is used for the decision.

One of the disadvantages Bell (2014) reminds of is that the decision trees can create complex models, resulting in overfitting. Overfitting means that the tree becomes too

large and does not classify new cases well, as explained in section 4.4.4. Tree pruning can be applied to resolve the problem as explained by Kabacoff (2015): In tree pruning, the tree with the lowest 10-fold cross-validation prediction error is chosen to avoid overfitting.

4.5.3 Random Forest

Another tree-based algorithm is the random forest. Random forest, interpreted in figure 10, is a classification and regression algorithm, which is an ensemble of uncorrelated decision trees. Random forest usually consists of tens or hundreds of decision trees and is applied if the training set is large. (Mueller & Massaron, 2016; Kamath & Kamat, 2016)

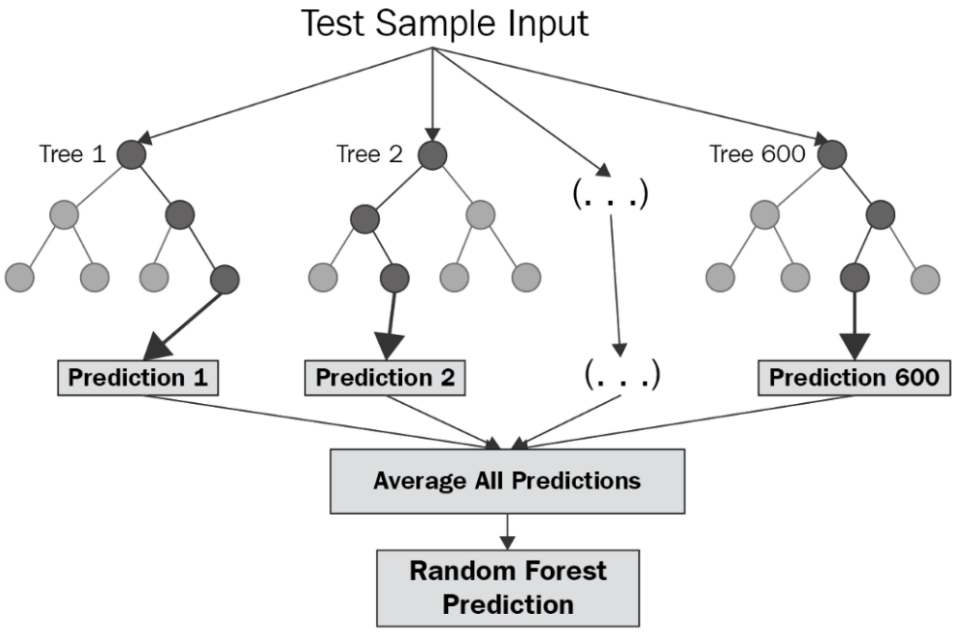

Figure 10. *Random forest*. (Chakure, 2019. Retrieved from Medium)

The random forest algorithm was created by Breiman and Cutler and according to Mueller & Massaron (2016), the purpose was to create an algorithm that is easy to use with little pre-processing and has few hyperparameters. The goal was to make the algorithm understandable and thus, the base is from decision trees. Random forest has features, such as bootstrapping, that are not used in decision trees.

Random forest uses bootstrapping with the training set. In bootstrapping, examples are sampled from a training set to create a new set. Thus, the examples are sampled multiple times. The trees in random forest are created with the bootstrapped examples and the best

splits are based on randomly picking features from the training set. Every tree created in the random forest is different from each other. The solution of the random forest model is taken with an average or a vote on new examples, both of which limit bias. In decision trees within the forest one feature might be dominant. As random forest consists of multiple trees, one tree that does not contain the dominant feature is able to find a different way to create branches and leaves. (Mueller & Massaron, 2016)

Mueller & Massaron (2016) point out that as random forest can contain hundreds of trees, it takes a lot of computational power and time to construct. The larger the forest is, the longer it takes to create. In addition, Murphy (2012) notes that a challenge with the algorithms using multiple trees is that they are not clearly interpretable compared to decision trees.

As mentioned above, random forest demands a vast amount of computational power and credit data sets are large. However, Brown & Mues (2012), Zhu et al. (2019), Tian et al. (2020), Yu (2020), Lessmann et al. (2015) and Wang et al. (2018) applied random forest in their loan default predictions. Zhu et al. (2019) accomplished most powerful results on random forest of the tree algorithms used. In addition, Wang et al. (2018) achieved highest AUC on their random forest model compared to logistic regression and the Cox model. As mentioned in section 3.1, Tian et al. (2020) study had an AUC of 0.96 and an accuracy of 88.96% for their random forest model. Based on the AUC, only gradient boosting tree outperformed their random forest model. Brown & Mues (2012) concluded that random forest performed well with an imbalanced data set. They set parameters for number of trees and the number of attributes used to grow the trees. In addition, Brown & Mues (2012) mention that they used 10-fold cross-validation in parameter tuning.

4.5.4 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is another tree-based algorithm which has been widely used in the last years. It has proved to be extremely effective with different machine learning challenges and as mentioned by Brownlee (2016a), it supports regression and classification problems. For example, Kaggle competitions have been dominated by methods utilizing XGBoost. Compared to other systems, XGBoost is more than ten times faster and scales to billions of examples (Chen & Guestrin, 2016).

In addition to Kaggle competitions, the research on loan default predictions has an increasing number of research which apply XGBoost. Odegua (2020), Xu et al. (2021) and Xia et al. (2017) applied XGBoost in their studies. Odegua (2020) studied loan default predictions with Python and purely on XGBoost. He mentions that he used 5-fold cross-validation to reduce bias and grid search to find the optimal parameters. Xu et al. (2021) and Xia et al. (2017) applied XGBoost to peer-to-peer lending which differs from the business-to-customer lending that this paper discusses. Both studies used k-fold cross-validation in hyperparameter tuning and Xia et al. (2017) utilized Bayesian hyper-parameter optimization to find the optimal parameters. They explain that grid search used by Odegua (2020) is a typical method but when a model has many hyperparameters, the number of possible combinations increase, which makes grid search inefficient.

XGBoost uses tree boosting, which according to Chen & Guestrin (2016) is a highly effective method. Gandhi (2018a) explains boosting as a method of turning weak learners into strong learners. A weak learner could have an error rate of 0.5, meaning that the predictive power is a coin toss. A strong learner's error rate is 0.0, making no errors. A group of weak learners is combined and voted on, making the group of weak learners into strong learners. Brownlee (2016a) elaborates that boosting is a technique which adds new models step by step to correct the errors made by previous models until the model cannot be improved.

In gradient tree boosting, the model uses weighted sums of multiple models. Gradient boosting creates new models by predicting residuals or errors from previous models with the gradient descent algorithm. The algorithm is used to minimize loss when adding new models. To make the final prediction, the residuals and errors are added together to select the best examples. (Brownlee, 2016a; Mueller & Massaron, 2016)

Gandhi (2018b) mentions that XGBoost is similar to gradient tree boosting, but XGBoost has a few more features, which make it so powerful and fast. These features are clever penalisation of trees, shrinking of leaf nodes, Newton boosting and a randomisation parameter. The trees can have different number of terminal leaves and the trees calculated with less evidence, are shrunk. The randomisation parameter can be used to reduce correlation between the trees, which makes for better classifications.

5 EMPIRICAL STUDY

After the literature review, the fifth chapter explains the empirical study and how the loan default prediction was conducted. The empirical study follows the model building steps explained in chapter 4. Machine learning algorithms used in the research are logistic regression, decision tree, random forest and XGBoost. First, the research method and data gathering are explained, followed by data description. Furthermore, various data pre-processing methods are explained in detail and model creation with prediction results of each model are displayed. Results analysis concludes chapter 5.

5.1 Method

The empirical study was conducted using R programming with an open banking data set retrieved from Kaggle.com. The data set's uploader has not given specific information about the source of the data. He mentioned that the data set is based on a real-life scenario, but it has been manipulated to be anonymous. The usability on the data set is 8.5, which is the best usability on credit data sets searched in Kaggle.

Finnish banks did not provide open data and the writer was not able to secure data from them due to bank secrecy laws. The purpose was to find a data set with similar variables which Finnish banks use in loan applications. Based on the writer's experience in loan granting, the data set used had the most similar variables and was therefore chosen.

5.2 Data Overview

The original data set from Kaggle consisted of three different Excel sheets. One sheet had application data from a customer, the second sheet had information about previous application and the last one contained columns description. Information about the previous application was not used in this research. The previous application did not provide any extra value to the application data according to the researcher's own experience on loan granting. The study was implemented with only the application data.

The data set consists of 307 511 observations and 122 different variables, which contain numerical and categorical values. Figure 11 displays the original data set. The data set

has two kinds of loans: Cash loans and revolving loans in “NAME_CONTRACT_TYPE” column. Cash loans are loans where a debtor has borrowed a certain amount of money from a bank and will pay back the loan according to loan terms. Mortgages and car loans are examples of cash loans. Revolving loans can also be described as flexi credits as the debtor has a personal line of credit, which he or she can use, payback and withdraw again. Credit cards are examples of such loans. In the data set, it is unknown what the debtor has purchased with the loan. Some might have bought a car and others might have bought a house. Some estimates can be derived from column “AMT_GOODS_PRICE”, which indicates the price of the good bought.

	Length	Mode
SK_ID_CURR	307511	numeric
TARGET	307511	numeric
NAME_CONTRACT_TYPE	307511	character
CODE_GENDER	307511	character
FLAG_OWN_CAR	307511	character
FLAG_OWN_REALTY	307511	character
CNT_CHILDREN	307511	numeric
AMT_INCOME_TOTAL	307511	numeric
AMT_CREDIT	307511	numeric
AMT_ANNUITY	307511	numeric
AMT_GOODS_PRICE	307511	numeric
NAME_TYPE_SUITE	307511	character
NAME_INCOME_TYPE	307511	character
NAME_EDUCATION_TYPE	307511	character
NAME_FAMILY_STATUS	307511	character
NAME_HOUSING_TYPE	307511	character
REGION_POPULATION_RELATIVE	307511	numeric
DAYS_BIRTH	307511	numeric
DAYS_EMPLOYED	307511	numeric
DAYS_REGISTRATION	307511	numeric
DAYS_ID_PUBLISH	307511	numeric
OWN_CAR_AGE	307511	numeric
FLAG_MOBIL	307511	numeric
FLAG_EMP_PHONE	307511	numeric
FLAG_WORK_PHONE	307511	numeric
FLAG_CONT_MOBILE	307511	numeric
FLAG_PHONE	307511	numeric
FLAG_EMAIL	307511	numeric
OCCUPATION_TYPE	307511	character
CNT_FAM_MEMBERS	307511	numeric
REGION_RATING_CLIENT	307511	numeric
REGION_RATING_CLIENT_W_CITY	307511	numeric
WEEKDAY_APPR_PROCESS_START	307511	character
HOUR_APPR_PROCESS_START	307511	numeric
REG_REGION_NOT_LIVE_REGION	307511	numeric
REG_REGION_NOT_WORK_REGION	307511	numeric
LIVE_REGION_NOT_WORK_REGION	307511	numeric
REG_CITY_NOT_LIVE_CITY	307511	numeric
REG_CITY_NOT_WORK_CITY	307511	numeric
LIVE_CITY_NOT_WORK_CITY	307511	numeric
ORGANIZATION_TYPE	307511	character
EXT_SOURCE_1	307511	numeric
EXT_SOURCE_2	307511	numeric
EXT_SOURCE_3	307511	numeric
APARTMENTS_AVG	307511	numeric
BASEMENTAREA_AVG	307511	numeric
YEARS_BEGINEXPLUATATION_AVG	307511	numeric
YEARS_BUILD_AVG	307511	numeric
COMMONAREA_AVG	307511	numeric
ELEVATORS_AVG	307511	numeric
ENTRANCES_AVG	307511	numeric
FLOORSMAX_AVG	307511	numeric
FLOORSMIN_AVG	307511	numeric
LANDAREA_AVG	307511	numeric
LIVINGAPARTMENTS_AVG	307511	numeric
LIVINGAREA_AVG	307511	numeric
AMT_REQ_CREDIT_BUREAU_HOUR	307511	numeric
AMT_REQ_CREDIT_BUREAU_DAY	307511	numeric
AMT_REQ_CREDIT_BUREAU_WEEK	307511	numeric
AMT_REQ_CREDIT_BUREAU_MON	307511	numeric
AMT_REQ_CREDIT_BUREAU_QRT	307511	numeric
AMT_REQ_CREDIT_BUREAU_YEAR	307511	numeric
NONLIVINGAPARTMENTS_AVG	307511	numeric
NONLIVINGAREA_AVG	307511	numeric
APARTMENTS_MODE	307511	numeric
BASEMENTAREA_MODE	307511	numeric
YEARS_BEGINEXPLUATATION_MODE	307511	numeric
YEARS_BUILD_MODE	307511	numeric
COMMONAREA_MODE	307511	numeric
ELEVATORS_MODE	307511	numeric
ENTRANCES_MODE	307511	numeric
FLOORSMAX_MODE	307511	numeric
FLOORSMIN_MODE	307511	numeric
LANDAREA_MODE	307511	numeric
LIVINGAPARTMENTS_MODE	307511	numeric
LIVINGAREA_MODE	307511	numeric
NONLIVINGAPARTMENTS_MODE	307511	numeric
NONLIVINGAREA_MODE	307511	numeric
APARTMENTS_MEDI	307511	numeric
BASEMENTAREA_MEDI	307511	numeric
YEARS_BEGINEXPLUATATION_MEDI	307511	numeric
YEARS_BUILD_MEDI	307511	numeric
COMMONAREA_MEDI	307511	numeric
ELEVATORS_MEDI	307511	numeric
ENTRANCES_MEDI	307511	numeric
FLOORSMAX_MEDI	307511	numeric
FLOORSMIN_MEDI	307511	numeric
LANDAREA_MEDI	307511	numeric
LIVINGAPARTMENTS_MEDI	307511	numeric
LIVINGAREA_MEDI	307511	numeric
NONLIVINGAPARTMENTS_MEDI	307511	numeric
NONLIVINGAREA_MEDI	307511	numeric
FONDKAPREMONT_MODE	307511	character
HOUSETYPE_MODE	307511	character
TOTALAREA_MODE	307511	numeric
WALLSMATERIAL_MODE	307511	character
EMERGENCYSTATE_MODE	307511	character
OBS_30_CNT_SOCIAL_CIRCLE	307511	numeric
DEF_30_CNT_SOCIAL_CIRCLE	307511	numeric
OBS_60_CNT_SOCIAL_CIRCLE	307511	numeric
DEF_60_CNT_SOCIAL_CIRCLE	307511	numeric
DAYS_LAST_PHONE_CHANGE	307511	numeric
FLAG_DOCUMENT_2	307511	numeric
FLAG_DOCUMENT_3	307511	numeric
FLAG_DOCUMENT_4	307511	numeric
FLAG_DOCUMENT_5	307511	numeric
FLAG_DOCUMENT_6	307511	numeric
FLAG_DOCUMENT_7	307511	numeric
FLAG_DOCUMENT_8	307511	numeric
FLAG_DOCUMENT_9	307511	numeric
FLAG_DOCUMENT_10	307511	numeric
FLAG_DOCUMENT_11	307511	numeric
FLAG_DOCUMENT_12	307511	numeric
FLAG_DOCUMENT_13	307511	numeric
FLAG_DOCUMENT_14	307511	numeric
FLAG_DOCUMENT_15	307511	numeric
FLAG_DOCUMENT_16	307511	numeric
FLAG_DOCUMENT_17	307511	numeric
FLAG_DOCUMENT_18	307511	numeric
FLAG_DOCUMENT_19	307511	numeric
FLAG_DOCUMENT_20	307511	numeric
FLAG_DOCUMENT_21	307511	numeric

Figure 11. *All variables in the used data set*

Variable “TARGET” is a binary variable as it has two possible values: 0 for a non-defaulted loan and 1 for a defaulted loan. The “TARGET” variable was used as the dependent variable. The proportion of non-defaulted loans is 91,9% and defaulted loans 8,1%, which indicates that the data set is heavily imbalanced as they generally are in real-life loan default problems. The imbalance of the “TARGET” variable is displayed in figure 12. Following chapters demonstrate how missing values, and the imbalance were treated.

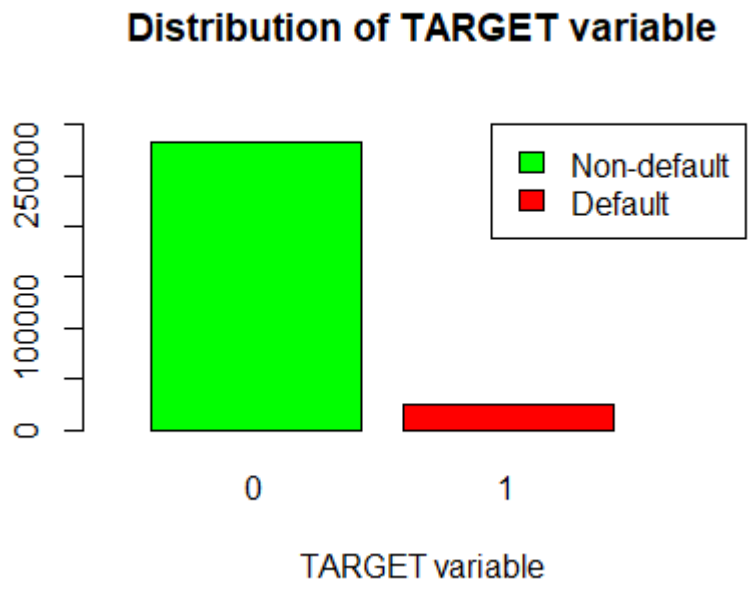

Figure 12. *Distribution of TARGET variable*

5.3 Data Cleaning

5.3.1 Variable Selection

As mentioned above, the data set consisted of 122 variables. The first step was to choose the variables that could have predictive value. First in data cleaning, the columns that had over 30% of missing values and variables that were not fully understood were deleted from the data set. The variables deleted due to containing more than 30 percentage of missing values are displayed on figure 13. Figure 13 displays variable name and the missing value percentage.

```
$OWN_CAR_AGE
[1] 0.6599081
$EXT_SOURCE_1
[1] 0.5638107
$APARTMENTS_AVG
[1] 0.5074973
$BASEMENTAREA_AVG
[1] 0.5851596
$YEARS_BEGINEXPLUATATION_AVG
[1] 0.4878102
$YEARS_BUILD_AVG
[1] 0.6649778
$COMMONAREA_AVG
[1] 0.698723
$ELEVATORS_AVG
[1] 0.5329598
$ENTRANCES_AVG
[1] 0.5034877
$FLOORSMAX_AVG
[1] 0.4976082
$FLOORSMIN_AVG
[1] 0.6784863
$LANDAREA_AVG
[1] 0.5937674
$LIVINGAPARTMENTS_AVG
[1] 0.6835495
$LIVINGAREA_AVG
[1] 0.5019333
$NONLIVINGAPARTMENTS_AVG
[1] 0.6943296
$NONLIVINGAREA_AVG
[1] 0.5517916
$APARTMENTS_MODE
[1] 0.5074973
$BASEMENTAREA_MODE
[1] 0.5851596
$YEARS_BEGINEXPLUATATION_MODE
[1] 0.4878102
$YEARS_BUILD_MODE
[1] 0.6649778
$COMMONAREA_MODE
[1] 0.698723
$ELEVATORS_MODE
[1] 0.5329598
$ENTRANCES_MODE
[1] 0.5034877
$FLOORSMAX_MODE
[1] 0.4976082
$LANDAREA_MODE
[1] 0.5937674
$LIVINGAPARTMENTS_MODE
[1] 0.6835495
$LIVINGAREA_MODE
[1] 0.5019333
$NONLIVINGAPARTMENTS_MODE
[1] 0.6943296
$NONLIVINGAREA_MODE
[1] 0.5517916
$APARTMENTS_MEDI
[1] 0.5074973
$BASEMENTAREA_MEDI
[1] 0.5851596
$YEARS_BEGINEXPLUATATION_MEDI
[1] 0.4878102
$YEARS_BUILD_MEDI
[1] 0.6649778
$COMMONAREA_MEDI
[1] 0.698723
$ELEVATORS_MEDI
[1] 0.5329598
$ENTRANCES_MEDI
[1] 0.5034877
$FLOORSMAX_MEDI
[1] 0.4976082
$FLOORSMIN_MEDI
[1] 0.6784863
$LANDAREA_MEDI
[1] 0.5937674
$LIVINGAPARTMENTS_MEDI
[1] 0.6835495
$LIVINGAREA_MEDI
[1] 0.5019333
$NONLIVINGAPARTMENTS_MEDI
[1] 0.6943296
$NONLIVINGAREA_MEDI
[1] 0.5517916
```

Figure 13. *Variables with more than 30% of missing values*

The variables which were not fully understood or described are displayed in figure 14. In addition, columns “SK_ID_CURR” and “FLAG_MOBIL” were deleted. “SK_ID_CURR” contained only the customer’s ID number and did not have any predictive value. “FLAG_MOBIL” had only one observation of a person not giving his or her phone number. As every other customer had a phone number, “FLAG_MOBIL” did not contain any predictive value.

	Length	Mode
EXT_SOURCE_2	307511	numeric
EXT_SOURCE_3	307511	numeric
AMT_REQ_CREDIT_BUREAU_HOUR	307511	numeric
AMT_REQ_CREDIT_BUREAU_DAY	307511	numeric
AMT_REQ_CREDIT_BUREAU_WEEK	307511	numeric
AMT_REQ_CREDIT_BUREAU_MON	307511	numeric
AMT_REQ_CREDIT_BUREAU_QRT	307511	numeric
AMT_REQ_CREDIT_BUREAU_YEAR	307511	numeric
FONDKAPREMONT_MODE	307511	character
HOUSETYPE_MODE	307511	character
WALLSMATERIAL_MODE	307511	character
EMERGENCYSTATE_MODE	307511	character
OBS_30_CNT_SOCIAL_CIRCLE	307511	numeric
DEF_30_CNT_SOCIAL_CIRCLE	307511	numeric
OBS_60_CNT_SOCIAL_CIRCLE	307511	numeric
DEF_60_CNT_SOCIAL_CIRCLE	307511	numeric
FLAG_DOCUMENT_2	307511	numeric
FLAG_DOCUMENT_3	307511	numeric
FLAG_DOCUMENT_4	307511	numeric
FLAG_DOCUMENT_5	307511	numeric
FLAG_DOCUMENT_6	307511	numeric
FLAG_DOCUMENT_7	307511	numeric
FLAG_DOCUMENT_8	307511	numeric
FLAG_DOCUMENT_9	307511	numeric
FLAG_DOCUMENT_10	307511	numeric
FLAG_DOCUMENT_11	307511	numeric
FLAG_DOCUMENT_12	307511	numeric
FLAG_DOCUMENT_13	307511	numeric
FLAG_DOCUMENT_14	307511	numeric
FLAG_DOCUMENT_15	307511	numeric
FLAG_DOCUMENT_16	307511	numeric
FLAG_DOCUMENT_17	307511	numeric
FLAG_DOCUMENT_18	307511	numeric
FLAG_DOCUMENT_19	307511	numeric
FLAG_DOCUMENT_20	307511	numeric
FLAG_DOCUMENT_21	307511	numeric

Figure 14. *Deleted values which were not described*

5.3.2 New Columns

The columns in figure 15 were proposed as negative numbers in the data set. Column “DAYS_BIRTH” shows how many days ago the applicant was born and “DAYS_EMPLOYED” displays how many days before the application he or she started current employment. Figure 15 indicates that variable “DAYS_EMPLOYED” consists of positive and negative values.

After further data analysis, the only positive values in “DAYS_EMPLOYED” were values of 365243. Based on data analysis and Kaggle conversations, the value 365243 is

a default value given to applicants who were not employed (pensioners) at the time of their application. Therefore, as other values were negative and the variable indicates how many days one has been in current employment, the values were turned to positive. Handling the value of 365243 will be introduced in section 5.3.4. Next, the variables in figure 15 were turned into positive numbers, which is demonstrated in figure 16.

```
  DAYS_BIRTH     DAYS_EMPLOYED    DAYS_REGISTRATION DAYS_ID_PUBLISH
Min.   :-25229   Min.   :-17912   Min.   :-24672    Min.   :-7197
1st Qu.:-19682   1st Qu.: -2760   1st Qu.: -7480    1st Qu.:-4299
Median :-15750   Median : -1213   Median : -4504    Median :-3254
Mean   :-16037   Mean   : 63815   Mean   : -4986    Mean   :-2994
3rd Qu.:-12413   3rd Qu.:  -289   3rd Qu.: -2010    3rd Qu.:-1720
Max.   : -7489   Max.   :365243   Max.   :     0    Max.   :    0

DAYS_LAST_PHONE_CHANGE
Min.   :-4292.0
1st Qu.:-1570.0
Median : -757.0
Mean   : -962.9
3rd Qu.: -274.0
Max.   :    0.0
NA's   :1
```

Figure 15. *Demonstration of negative values*

```
  DAYS_BIRTH    DAYS_EMPLOYED    DAYS_REGISTRATION DAYS_ID_PUBLISH
Min.   : 7489   Min.   :     0   Min.   :    0     Min.   :   0
1st Qu.:12413   1st Qu.:   933   1st Qu.: 2010     1st Qu.:1720
Median :15750   Median :  2219   Median : 4504     Median :3254
Mean   :16037   Mean   : 67725   Mean   : 4986     Mean   :2994
3rd Qu.:19682   3rd Qu.:  5707   3rd Qu.: 7480     3rd Qu.:4299
Max.   :25229   Max.   :365243   Max.   :24672     Max.   :7197

DAYS_LAST_PHONE_CHANGE
Min.   :   0.0
1st Qu.: 274.0
Median : 757.0
Mean   : 962.9
3rd Qu.:1570.0
Max.   :4292.0
NA's   :1
```

Figure 16. *After changing negative values into positive*

Furthermore, the values in figure 16 were turned to “Age”, “YearsWorked”, “Years_since_registration”, “Years_since_ID_publ” and “Years_since_Phone_Change” and the variables in figures 15 & 16 were deleted. Figure 17 displays the new variables.

To clarify, the variables tell how old a person is in years, how many years a person has been working in his or her current employment, how many years before the application a person changed his or her registration and how many years before sending the application was the ID card and phone number changed.

```
      Age          YearsWorked     Years_since_registration Years_since_ID_publ
 Min.   :21.00   Min.   :   0.0   Min.   : 0.000           Min.   : 0.000
 1st Qu.:34.00   1st Qu.:   3.0   1st Qu.: 5.507           1st Qu.: 4.712
 Median :43.00   Median :   6.0   Median :12.340           Median : 8.915
 Mean   :43.94   Mean   : 185.6   Mean   :13.661           Mean   : 8.203
 3rd Qu.:54.00   3rd Qu.:  16.0   3rd Qu.:20.492           3rd Qu.:11.778
 Max.   :69.00   Max.   :1001.0   Max.   :67.595           Max.   :19.718

 Years_since_Phone_Change
 Min.   : 0.0000
 1st Qu.: 0.7507
 Median : 2.0740
 Mean   : 2.6380
 3rd Qu.: 4.3014
 Max.   :11.7589
 NA's   :1
```

Figure 17. *After changing days into years*

The columns in figure 17 and columns “CNT_CHILDREN_RANK” and “FAM_MEMBERS_RANK” in figure 18 were created for more predictive value. Number of children and number of family members were turned into ranks ranging from 1 to 5. In these ranks, 1 means no children or other family members, 2 means one or two children or family members, 3 stands for three or four children or family members, 4 for five or six and 5 means seven or more children or other family members. The original columns “CNT_CHILDREN” and “CNT_FAM_MEMBERS” were deleted from the data set after creating new variables.

```
  CNT_CHILDREN       CNT_FAM_MEMBERS     CNT_CHILDREN_RANK FAM_MEMBERS_RANK
 Min.   : 0.0000   Min.   : 1.000     Min.   :1.000     Min.   :2.000
 1st Qu.: 0.0000   1st Qu.: 2.000     1st Qu.:1.000     1st Qu.:2.000
 Median : 0.0000   Median : 2.000     Median :1.000     Median :2.000
 Mean   : 0.4171   Mean   : 2.153     Mean   :1.314     Mean   :2.278
 3rd Qu.: 1.0000   3rd Qu.: 3.000     3rd Qu.:2.000     3rd Qu.:3.000
 Max.   :19.0000   Max.   :20.000     Max.   :5.000     Max.   :5.000
                   NA's   :2
```

Figure 18. *CNT_CHILDREN and CNT_FAM_MEMBERS before and after changed to ranks.*

In addition, reclassifying character values was performed to optimize the learning algorithms. The character columns consisted of many unique values. Variable “ORGANIZATION_TYPE” consisted of 58 unique values. “ORGANIZATION_TYPE”

had 13 unique values for industry, named “Industry: Type 1, Industry: Type 2, Industry: Type 3” etc., seven different values for “Trade” and four different types for “Transport”. These similar values were combined in order to increase their predictive values and to decrease the number of unique values. The outcome of “ORGANIZATION_TYPE” after combining values is displayed in figure 19.

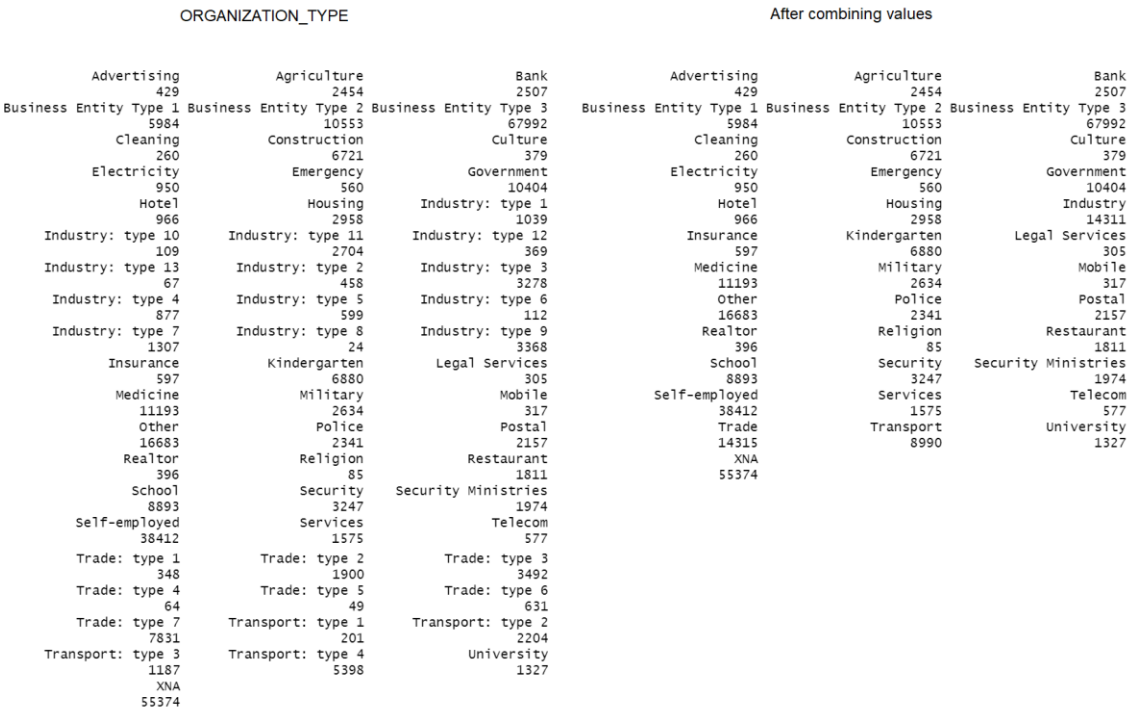

Figure 19. ORGANIZATION_TYPE before and after combining values

The following columns were created as they are important in loan granting in Finnish banks and in order to improve predictive power. The data set has a variable “AMT_GOODS_PRICE” which illustrates the price of the good purchased with the credit. “AMT_CREDIT” shows how much credit was granted in the application. To create the new column called “LTV” (Loan-To-Value), “AMT_GOODS_PRICE” was divided by “AMT_CREDIT”. The result is the loan-to-value ratio that was introduced in section 2.1.

Column “Payment_perc” was created to present how many percentages of a customer’s yearly income is appointed to loan payments. The variable was created by dividing variable “AMT_ANNUITY” by “AMT_INCOME_TOTAL”. “AMT_ANNUITY” indicates yearly payment on the loan and “AMT_INCOME_TOTAL” illustrates the

customer's yearly income. "LTV" and "Payment_perc" columns are presented in figure 20.

```
      LTV          Payment_perc
Min.   :0.1700   Min.   :0.0000
1st Qu.:0.8300   1st Qu.:0.1100
Median :0.8900   Median :0.1600
Mean   :0.8998   Mean   :0.1809
3rd Qu.:1.0000   3rd Qu.:0.2300
Max.   :6.6700   Max.   :1.8800
NA's   :278      NA's   :12
```

Figure 20. *New, created columns*

Figure 21 displays the data after variables were deleted and new ones were created. The number of variables decreased from the original 122 to 41 variables, while the number of observations remained the same as in the original data set. The number of observations will decrease in the next section where missing values handling is executed.

	Length	Mode
TARGET	307511	numeric
NAME_CONTRACT_TYPE	307511	character
CODE_GENDER	307511	character
FLAG_OWN_CAR	307511	character
FLAG_OWN_REALTY	307511	character
AMT_INCOME_TOTAL	307511	numeric
AMT_CREDIT	307511	numeric
AMT_ANNUITY	307511	numeric
AMT_GOODS_PRICE	307511	numeric
NAME_TYPE_SUITE	307511	character
NAME_INCOME_TYPE	307511	character
NAME_EDUCATION_TYPE	307511	character
NAME_FAMILY_STATUS	307511	character
NAME_HOUSING_TYPE	307511	character
REGION_POPULATION_RELATIVE	307511	numeric
FLAG_EMP_PHONE	307511	numeric
FLAG_WORK_PHONE	307511	numeric
FLAG_CONT_MOBILE	307511	numeric
FLAG_PHONE	307511	numeric
FLAG_EMAIL	307511	numeric
OCCUPATION_TYPE	307511	character
REGION_RATING_CLIENT	307511	numeric
REGION_RATING_CLIENT_W_CITY	307511	numeric
WEEKDAY_APPR_PROCESS_START	307511	character
HOUR_APPR_PROCESS_START	307511	numeric
REG_REGION_NOT_LIVE_REGION	307511	numeric
REG_REGION_NOT_WORK_REGION	307511	numeric
LIVE_REGION_NOT_WORK_REGION	307511	numeric
REG_CITY_NOT_LIVE_CITY	307511	numeric
REG_CITY_NOT_WORK_CITY	307511	numeric
LIVE_CITY_NOT_WORK_CITY	307511	numeric
ORGANIZATION_TYPE	307511	character
CNT_CHILDREN_RANK	307511	numeric
FAM_MEMBERS_RANK	307511	numeric
Age	307511	numeric
YearsWorked	307511	numeric
LTV	307511	numeric
Payment_perc	307511	numeric
Years_since_registration	307511	numeric
Years_since_ID_publ	307511	numeric
Years_since_Phone_Change	307511	numeric

Figure 21. *Data set after deleting columns and creating new columns*

5.3.3 Missing Values

The existing variables still had missing values, which were displayed in different ways. Some missing values were displayed as “XNA”, some with a null or blank value and in some character variables with “unknown”. The following sections present how missing values in numerical and categorical columns were treated.

5.3.3.1 Character Variables

12 out of the 41 variables in figure 21 are character variables. Character variables "CODE_GENDER", "NAME_TYPE_SUITE", "NAME_FAMILY_STATUS", "OCCUPATION_TYPE" and "ORGANIZATION_TYPE" had missing values and as mentioned, they were displayed in different ways. The number of missing values was wide ranging from 96 391 blank values in "OCCUPATION_TYPE" to two "unknowns" in "NAME_FAMILY_STATUS", which are displayed in figure 22.

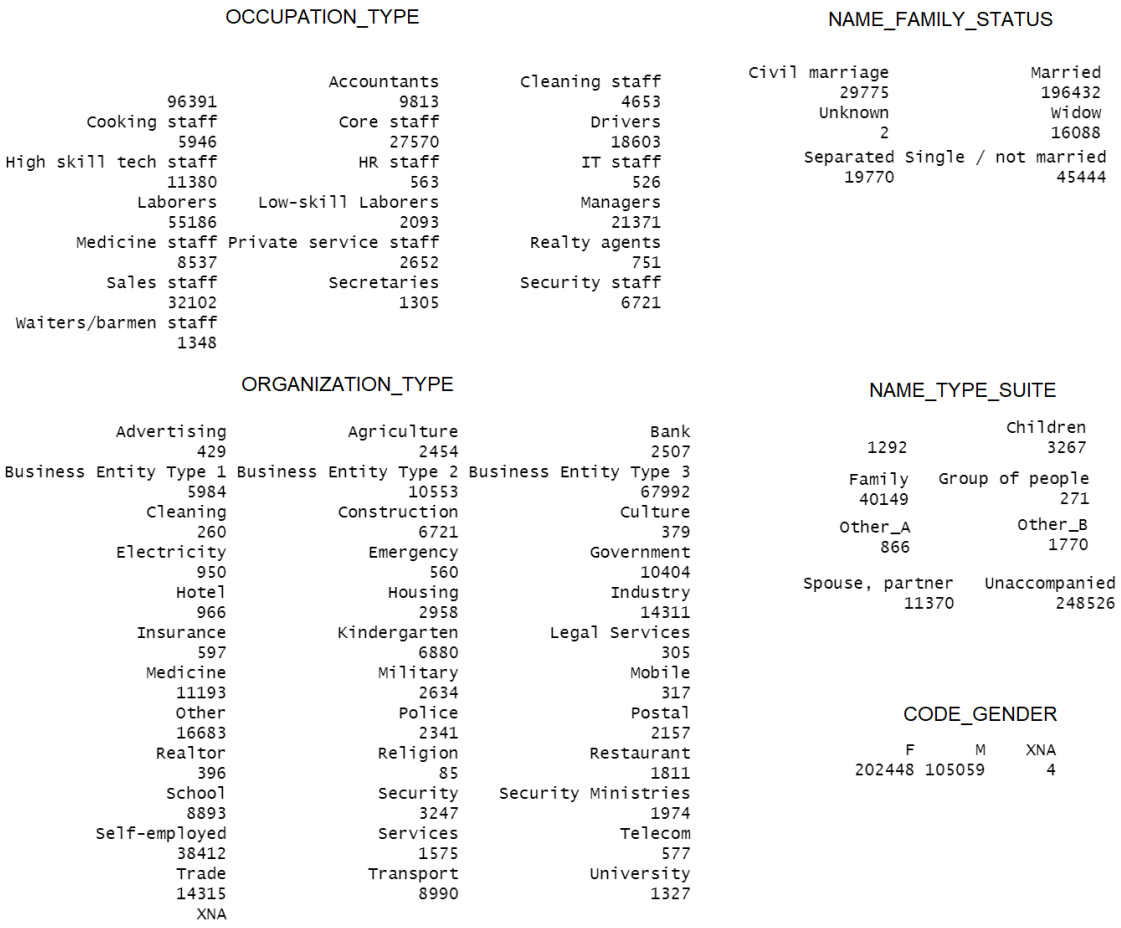

Figure 22. *OCCUPATION_TYPE, NAME_FAMILY_STATUS, ORGANIZATION_TYPE, NAME_TYPE_SUITE and CODE_GENDER before handling missing values*

The first step was to change null or blank values into NA values. Then, NA's were changed into the most frequent character in the column. The same was executed for "XNA" and "unknown" values separately to each column. For example, in the "OCCUPATION_TYPE" variable, the blank values were first changed into NA values.

These values were then assigned together with the most frequent value in the column, which in “OCCUPATION_TYPE” was laborers. The result is displayed in figure 23.

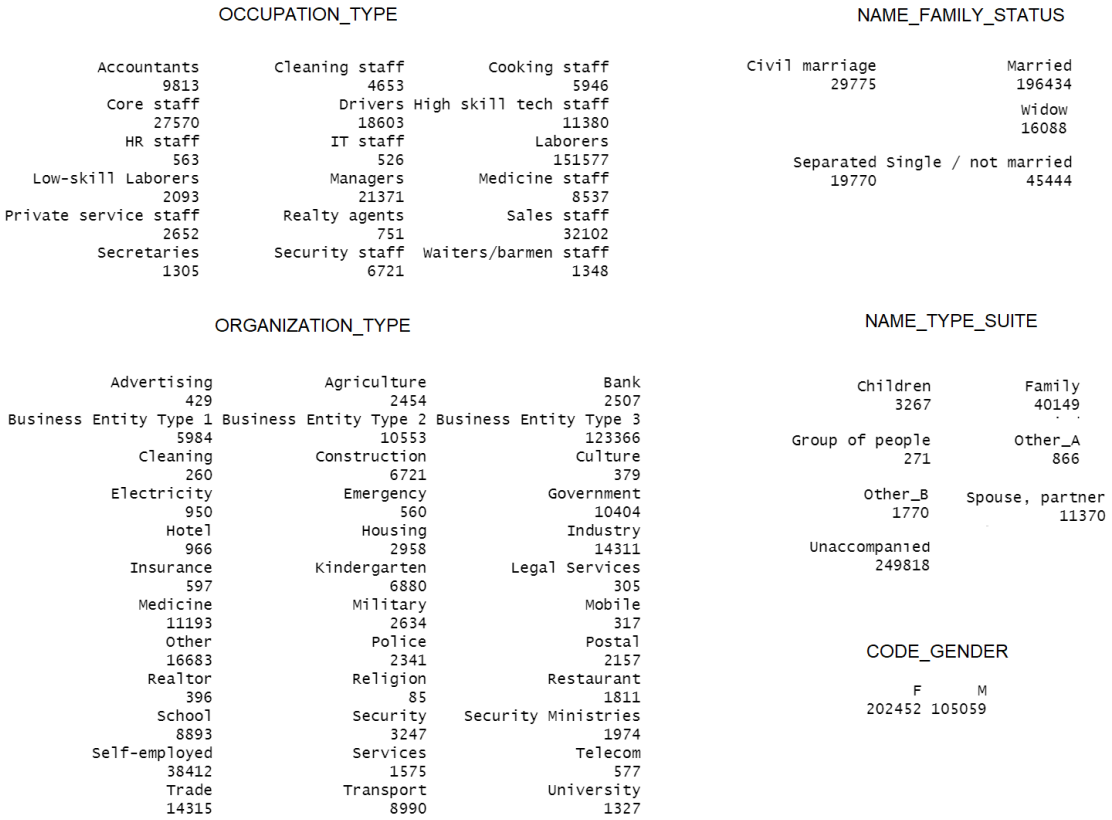

Figure 23. *OCCUPATION_TYPE, NAME_FAMILY_STATUS, ORGANIZATION_TYPE, NAME_TYPE_SUITE and CODE_GENDER after handling missing values*

5.3.3.2 Numerical Values

After creating new columns, the data set had 29 numeric variables. Out of these variables “AMT_ANNUITY” had 12 missing values, “AMT_GOODS_PRICE” 278 missing values, “Years_since_Phone_Change” one missing value, “LTV” 278 missing values and “Payment_perc” had 12 missing values. The variables “AMT_INCOME_TOTAL”, “AMT_CREDIT”, “REGION_POPULATION_RELATIVE”, “HOUR_APPR_PROCESS_START”, “Age”, “YearsWorked”, “Years_since_registration” and “Years_since_ID_publ” had no missing values and were kept as numerical values.

The other 16 numeric variables had no missing values, and were treated as categorical variables, including the “TARGET” variable indicating if a person has defaulted his or her loan. The following variables were turned into factors to exclude them from outlier

detection: “TARGET”, “FLAG_EMP_PHONE”, “FLAG_WORK_PHONE”, “FLAG_CONT_MOBILE”, “FLAG_PHONE”, “FLAG_EMAIL”, “REGION_RATING_CLIENT”, “REGION_RATING_CLIENT_W_CITY”, “REG_REGION_NOT_WORK_REGION”, “REG_REGION_NOT_LIVE_REGION”, “REG_CITY_NOT_LIVE_CITY”, “REG_CITY_NOT_WORK_CITY”, “LIVE_CITY_NOT_WORK_CITY”, “LIVE_REGION_NOT_WORK_REGION”, “CNT_CHIDREN_RANK” and “FAM_MEMBERS_RANK”.

As missing values in numerical values were only 0,19 % of all observations, it was decided that the missing values will be deleted from the data set. More data from the numerical variables will be deleted when handling outliers and the next section introduces the procedure.

5.3.4 Outliers

Visualization showed that variables “AMT_INCOME_TOTAL”, “AMT_CREDIT”, “AMT_ANNUITY”, “AMT_GOODS_PRICE”, “LTV” and “Payment_perc” had outliers that might have an effect on predictions. Visualization is displayed in figure 24. In addition, the boxplot in figure 26 indicated outliers in variable “YearsWorked” (“DAYS_EMPLOYED” before data cleaning process). However, as mentioned in section 5.3.2 “YearsWorked” had various values of 365243. Handling of these values is explained below. In order to detect outliers, the observations that were three standard deviations away from the mean were assigned as outliers and changed into NA’s. Changing values three standard deviations away from the mean into NA’s, resulted in 20 231 missing values which were deleted from the data set. Figure 25 displays the variables after deleting outliers.

“

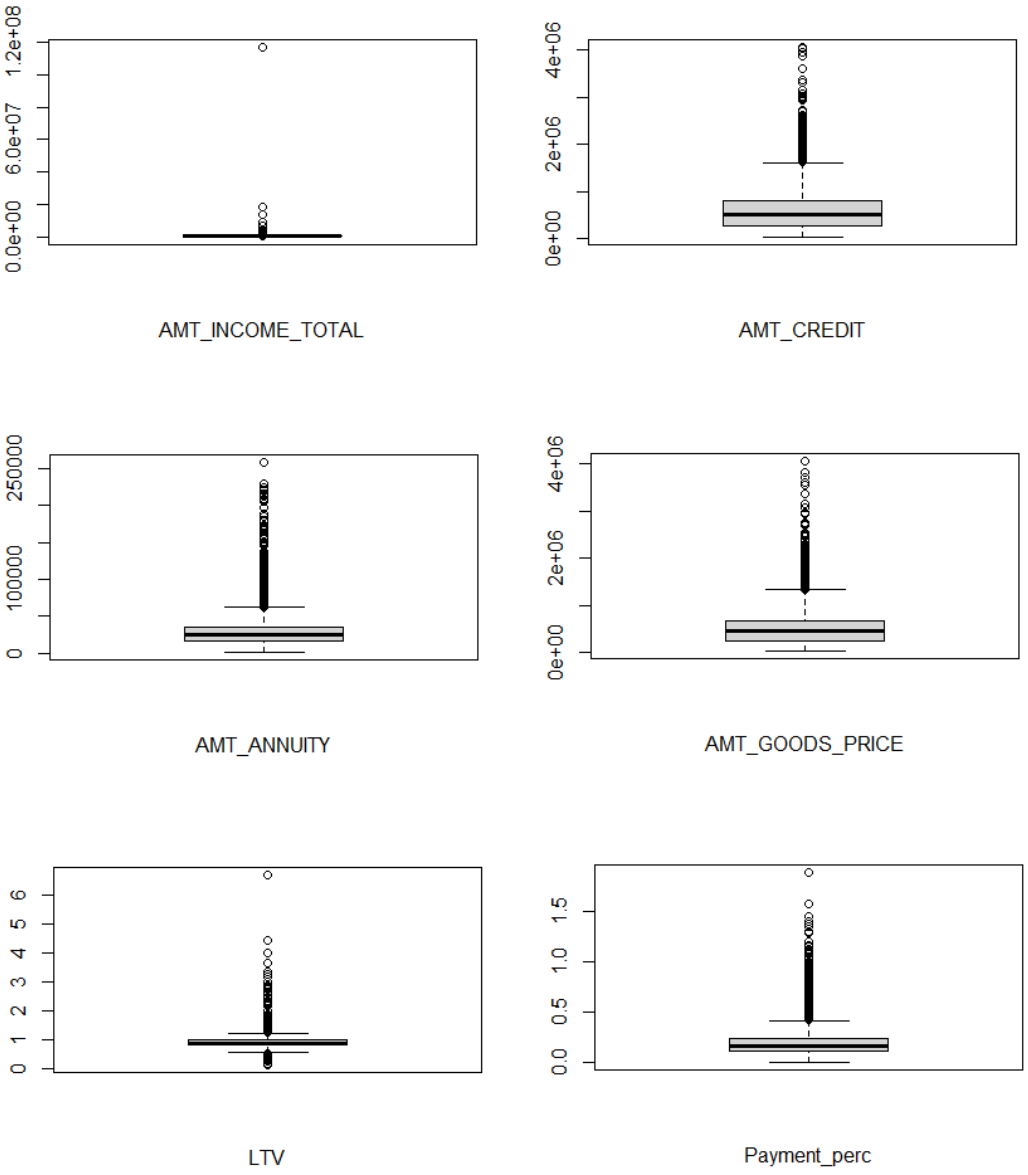

Figure 24. *Detecting outliers with boxplot*

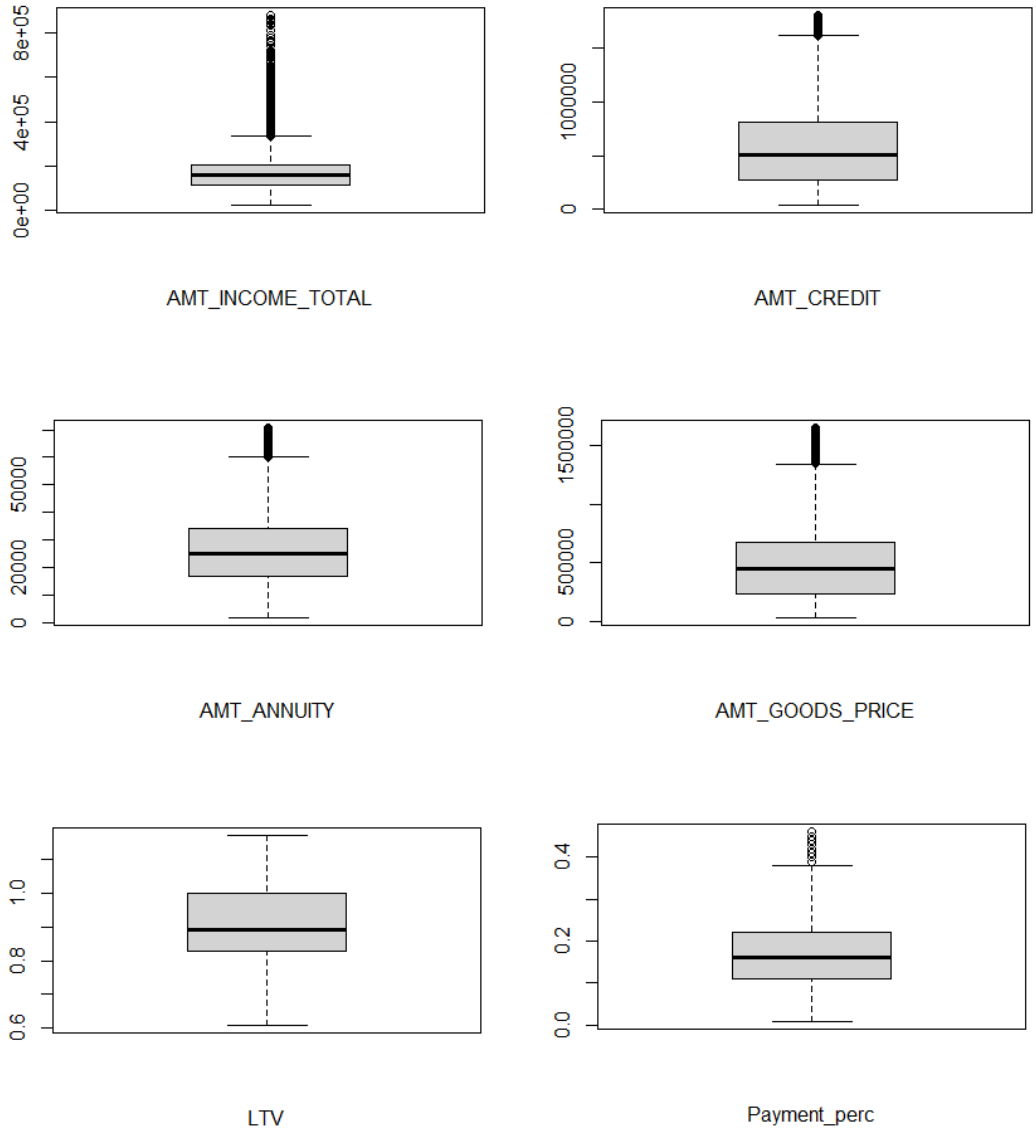

Figure 25. *Boxplots after removing outliers*

Variable "YearsWorked" demonstrates how many years one has been working in current employment. As mentioned in section 5.3.2, the variable had 51 232 observations indicating that one had been employed for 365243 days (1001 years), which was a default value for pensioners. Figure 26 displays the values in a boxplot. In order to keep the variable in the data set and for it to have predictive value, it was decided that these 51 232 observations will be deleted. Imputing the value with another value, such as 0 was not possible as 0 in the column implied unemployment.

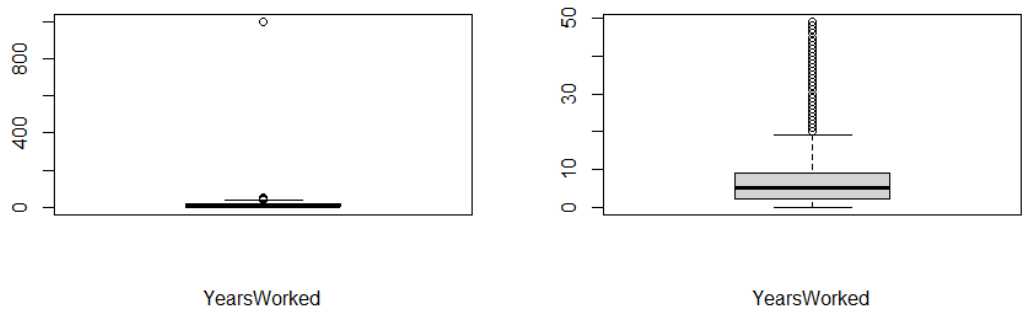

Figure 26. "*YearsWorked" before and after deleting values 1001*

Before data cleaning, the data set had 307 511 observations and 122 variables with 91,9% of non-defaulted loans and 8,1% of defaulted loans. After adding new variables and data cleaning, the number of observations was 236 048 and 41 variables with 91,1% of non-defaulted loans and 8,9% of defaulted loans. Approximately 23% of the observations were deleted from the data set. The percentage of deleted observations was rather high, but out of 71 463 deleted observations, 51 232 observations were values from the variable "YearsWorked". As explained above, the variable displayed a given default value (365243 days) for pensioners. The default rate for pensioners was low as 94,6% were non-defaulted loans and 5,4% defaulted loans. These values were deleted from the data set for simplicity. A new csv-file was created from the data set, and it was used to build the prediction models. Figure 27 displays the variables after data cleaning. After data cleaning, the data set was ready to be trained with machine learning models.

	Length	Class	Mode
TARGET	236048	factor	numeric
NAME_CONTRACT_TYPE	236048	factor	numeric
CODE_GENDER	236048	factor	numeric
FLAG_OWN_CAR	236048	factor	numeric
FLAG_OWN_REALTY	236048	factor	numeric
AMT_INCOME_TOTAL	236048	-none-	numeric
AMT_CREDIT	236048	-none-	numeric
AMT_ANNUITY	236048	-none-	numeric
AMT_GOODS_PRICE	236048	-none-	numeric
NAME_TYPE_SUITE	236048	factor	numeric
NAME_INCOME_TYPE	236048	factor	numeric
NAME_EDUCATION_TYPE	236048	factor	numeric
NAME_FAMILY_STATUS	236048	factor	numeric
NAME_HOUSING_TYPE	236048	factor	numeric
REGION_POPULATION_RELATIVE	236048	-none-	numeric
FLAG_EMP_PHONE	236048	factor	numeric
FLAG_WORK_PHONE	236048	factor	numeric
FLAG_CONT_MOBILE	236048	factor	numeric
FLAG_PHONE	236048	factor	numeric
FLAG_EMAIL	236048	factor	numeric
OCCUPATION_TYPE	236048	factor	numeric
REGION_RATING_CLIENT	236048	factor	numeric
REGION_RATING_CLIENT_W_CITY	236048	factor	numeric
WEEKDAY_APPR_PROCESS_START	236048	factor	numeric
HOUR_APPR_PROCESS_START	236048	-none-	numeric
REG_REGION_NOT_LIVE_REGION	236048	factor	numeric
REG_REGION_NOT_WORK_REGION	236048	factor	numeric
LIVE_REGION_NOT_WORK_REGION	236048	factor	numeric
REG_CITY_NOT_LIVE_CITY	236048	factor	numeric
REG_CITY_NOT_WORK_CITY	236048	factor	numeric
LIVE_CITY_NOT_WORK_CITY	236048	factor	numeric
ORGANIZATION_TYPE	236048	factor	numeric
CNT_CHILDREN_RANK	236048	factor	numeric
FAM_MEMBERS_RANK	236048	factor	numeric
Age	236048	-none-	numeric
YearsWorked	236048	-none-	numeric
LTV	236048	-none-	numeric
Payment_perc	236048	-none-	numeric
Years_since_registration	236048	-none-	numeric
Years_since_ID_publ	236048	-none-	numeric
Years_since_Phone_Change	236048	-none-	numeric

Figure 27. *Variables used in making predictions*

5.4 Model Creation

The purpose of this section is to introduce how the prediction models were built. The four models used for predictions were logistic regression, decision tree, random forest and XGBoost. Logistic regression and classification tree were chosen as they are commonly used classification methods and offer benchmarks for the predictions. Random forest is

developed from classification trees and XGBoost has been dominating classification contests, so it was of interest to see how well they perform compared to benchmark models.

K-fold cross-validation was performed on all models to reduce bias. In addition, a test set is held out for final evaluation. 70% of the observations were assigned to the training set and 30% to the test set. The training set was cross-validated, and the test set was used for evaluating the model.

5.4.1 Logistic Regression

After splitting the data to training set and test set, the first step in creating the logistic regression model was to review collinearity. Collinearity can affect a logistic regression model and therefore, variables with high correlation should be removed. In the data set, variables "AMT_GOOD_PRICE" and "AMT_CREDIT" had almost a perfect correlation of 0.985. Therefore, it was decided that "AMT_CREDIT" would not be used in the logistic regression model. After correlation was revised, standardizing the training and test sets was next. As the numeric variables had many values, scaling was done to prevent one significant number having too much power in the prediction.

5.4.1.1 Generalized Linear Model function

The generalized linear model (glm) function was applied to every variable in the data set with "TARGET" variable being the dependent variable. Every variable was used as it was of interest to know which variables have significance to "TARGET". As "TARGET" variable was binary, the family used in the glm function was set to binomial and the scaled training set was used as the data.

```
Coefficients:
                                                Estimate Std. Error z value  Pr(>|z|)
(Intercept)                                    -4.073897   0.597153  -6.822  8.97e-12 ***
`NAME_CONTRACT_TYPERevolving loans`            -0.199527   0.039737  -5.021  5.13e-07 ***
CODE_GENDERM                                    0.310145   0.022320  13.895   < 2e-16 ***
FLAG_OWN_CARY                                  -0.304620   0.020427 -14.913   < 2e-16 ***
AMT_INCOME_TOTAL                                0.101196   0.018777   5.390  7.06e-08 ***
AMT_GOODS_PRICE                                -0.181718   0.014909 -12.189   < 2e-16 ***
AMT_ANNUITY                                    -0.041059   0.019956  -2.057  0.039643 *
NAME_INCOME_TYPEPensioner                      -7.229109  52.088299  -0.139  0.889620
`NAME_INCOME_TYPEState servant`                -0.124070   0.042242  -2.937  0.003313 **
NAME_INCOME_TYPEWorking                         0.125493   0.021219   5.914  3.33e-09 ***
`NAME_EDUCATION_TYPEHigher education`           0.554312   0.592443   0.936  0.349460
`NAME_EDUCATION_TYPEIncomplete higher`          0.758541   0.593824   1.277  0.201467
`NAME_EDUCATION_TYPELower secondary`            1.175585   0.596999   1.969  0.048935 *
`NAME_EDUCATION_TYPESecondary / secondary special`  1.017603   0.592191   1.718  0.085729 .
NAME_FAMILY_STATUSMarried                      -0.141298   0.028075  -5.033  4.83e-07 ***
NAME_FAMILY_STATUSSeparated                    -0.007745   0.045291  -0.171  0.864217
`NAME_FAMILY_STATUSSingle / not married`       -0.028265   0.034072  -0.830  0.406788
NAME_FAMILY_STATUSWidow                        -0.244341   0.069587  -3.511  0.000446 ***
REGION_POPULATION_RELATIVE                      0.018195   0.010352   1.758  0.078814 .
FLAG_WORK_PHONE1                                0.219185   0.022020   9.954   < 2e-16 ***
FLAG_PHONE1                                    -0.080357   0.022210  -3.618  0.000297 ***
`OCCUPATION_TYPECleaning staff`                 0.388775   0.085459   4.549  5.38e-06 ***
`OCCUPATION_TYPECooking staff`                  0.297024   0.079572   3.733  0.000189 ***
`OCCUPATION_TYPECore staff`                     0.088735   0.066261   1.339  0.180514
OCCUPATION_TYPEDrivers                          0.394481   0.068439   5.764  8.21e-09 ***
`OCCUPATION_TYPEHigh skill tech staff`          0.043084   0.076143   0.566  0.571510
`OCCUPATION_TYPEHR staff`                       0.560896   0.205611   2.728  0.006373 **
`OCCUPATION_TYPEIT staff`                      -0.051733   0.236435  -0.219  0.826802
OCCUPATION_TYPELaborers                         0.276914   0.061305   4.517  6.27e-06 ***
`OCCUPATION_TYPELow-skill Laborers`             0.564848   0.094649   5.968  2.40e-09 ***
OCCUPATION_TYPEManagers                         0.196332   0.069159   2.839  0.004528 **
`OCCUPATION_TYPEMedicine staff`                 0.111810   0.080521   1.389  0.164964
`OCCUPATION_TYPEPrivate service staff`         -0.034151   0.114219  -0.299  0.764943
`OCCUPATION_TYPERealty agents`                  0.229564   0.171817   1.336  0.181517
`OCCUPATION_TYPESales staff`                    0.260146   0.063409   4.103  4.08e-05 ***
OCCUPATION_TYPESecretaries                      0.226237   0.146094   1.549  0.121484
`OCCUPATION_TYPESecurity staff`                 0.370610   0.077815   4.763  1.91e-06 ***
`OCCUPATION_TYPEWaiters/barmen staff`           0.386214   0.119002   3.245  0.001173 **
REGION_RATING_CLIENT_W_CITY2                    0.426058   0.041887  10.172   < 2e-16 ***
REGION_RATING_CLIENT_W_CITY3                    0.757202   0.048253  15.692   < 2e-16 ***
WEEKDAY_APPR_PROCESS_STARTMONDAY               -0.029589   0.031082  -0.952  0.341102
WEEKDAY_APPR_PROCESS_STARTSATURDAY             -0.019765   0.034414  -0.574  0.565749
WEEKDAY_APPR_PROCESS_STARTSUNDAY               -0.034479   0.044114  -0.782  0.434460
WEEKDAY_APPR_PROCESS_STARTTHURSDAY              0.039859   0.030803   1.294  0.195665
WEEKDAY_APPR_PROCESS_STARTTUESDAY               0.064755   0.030176   2.146  0.031881 *
WEEKDAY_APPR_PROCESS_STARTWEDNESDAY             0.034942   0.030594   1.142  0.253403
HOUR_APPR_PROCESS_START                        -0.041756   0.009134  -4.571  4.85e-06 ***
REG_CITY_NOT_LIVE_CITY1                         0.200220   0.027108   7.386  1.51e-13 ***
CNT_CHILDREN_RANK2                              0.125214   0.045383   2.759  0.005797 **
CNT_CHILDREN_RANK3                              0.419278   0.250117   1.676  0.093674 .
CNT_CHILDREN_RANK4                             -8.283569  84.363871  -0.098  0.921783
CNT_CHILDREN_RANK5                             -9.088199  84.371418  -0.108  0.914221
FAM_MEMBERS_RANK3                              -0.143612   0.049276  -2.914  0.003563 **
FAM_MEMBERS_RANK4                              -0.383966   0.260326  -1.475  0.140228
FAM_MEMBERS_RANK5                               8.497057  84.364747   0.101  0.919774
Age                                            -0.135269   0.010815 -12.507   < 2e-16 ***
YearsWorked                                    -0.223826   0.012270 -18.242   < 2e-16 ***
LTV                                            -0.206558   0.009539 -21.654   < 2e-16 ***
Payment_perc                                    0.122647   0.019006   6.453  1.10e-10 ***
Years_since_registration                       -0.052969   0.009661  -5.483  4.19e-08 ***
Years_since_ID_publ                            -0.100998   0.009121 -11.073   < 2e-16 ***
Years_since_Phone_Change                       -0.142059   0.009750 -14.570   < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 99213  on 165233  degrees of freedom
Residual deviance: 94047  on 165173  degrees of freedom
AIC: 94169

Number of Fisher Scoring iterations: 9
```

Figure 28. *Variable significance in logistic regression*

After checking the significance of all variables compared to “TARGET”, the variables in figure 28 indicated significance. These variables were used in the logistic regression model. Variables “FLAG_OWN_REALTY”, “NAME_TYPE_SUITE”, “NAME_HOUSING_TYPE”, “FLAG_EMP_PHONE”, “FLAG_EMAIL”,

"REGION_RATING_CLIENT", "REG_REGION_NOT_WORK_REGION", "LIVE_CITY_NOT_WORK_CITY", and "ORGANIZATION_TYPE" did not display significance and were not used in the glm function.

5.4.1.2 Prediction

After detecting the significant variables, the next step was to create the prediction. The prediction was executed with the predict function. The significant variables were used as an "object" in the function. In addition, test set was used as "newdata" and the "type" was set to "probabilities". Prediction outcomes from the test set were used in creating the confusion matrix and the ROC-curve and finding the optimal threshold with Youden's index. The evaluation metrics will be discussed more and displayed in section 5.5.

5.4.2 Classification tree

The outcome variable "TARGET" was a categorical value, therefore the decision tree built was a classification tree. In creating the classification tree, numeric variables were not standardized. Standardised values do not change the outcome of the prediction in decision trees. Another difference to building a logistic regression model, oversampling and undersampling of the data was needed in classification tree model. As mentioned, the data was split 70% to training set and 30% to test set.

5.4.2.1 Oversampling and Undersampling

As the model was first created, the classification tree failed and did not create a tree. The reason was the imbalanced data set. Therefore, it was decided that oversampling and undersampling will be tested for the classification tree to increase predictive power.

Oversampling and undersampling were both executed on the training set. For "ovun.sample" function used, the "TARGET" variable was set as the predictor and all variables were used in the model. For oversampling, "method" was set to "over" and for undersampling to "under". Sampling was executed based on the ratio of "TARGET" variable. The "TARGET" variable had 150 528 non-defaulted loans and 14 706 defaulted loans. In oversampling, the purpose was to increase the number of defaulted loans to the same level as the non-defaulted loans. It was achieved by setting "ovun.sample" function to 301 056 which is two times 150 528. For undersampling, the aim was to decrease the number of non-defaulted loans to the level of defaulted loans. In this case, sampling was

set to two times the defaulted loans, 29 412. As a result, the number of non-defaulted loans and the number of defaulted loans was 14 796. After oversampling and undersampling, the classification tree model was created.

5.4.2.2 Model Creation and Tree Pruning

Before assigning the data set into the rpart function, the oversampled and undersampled training sets were cross-validated. The cross-validated oversampled and undersampled data sets were used as "data" to compare which had better results. The "TARGET" variable was used as the dependent variable and all variables introduced in section 5.3.4 as predictors. To limit overfitting, "maxdepth" in both functions was set to five.

The tree was pruned for increased model performance. Pruning was performed by printing a complexity parameter (CP) table, which displays the cross-validation error or the x-error. The CP value containing the lowest x-error value was used to prune the tree on both data sets. The CP table also shows the optimal number of splits based on the CP value.

5.4.2.3 Prediction

In classification tree, the prediction was executed separately to the oversampled and undersampled data. To the predict function, the pruned trees were used as the "object", test set as "newdata" and the "type" in the function was set to "prob". Type "prob" returns the classification probabilities. The confusion matrix and ROC-curve, which were created after the predictions will be introduced in the results section.

5.4.3 Random Forest

The model creation for random forest was similar to classification tree. The numerical variables were not standardized, and the character variables were set as factors. The data was split as they were in the previous models.

5.4.3.1 Undersampling

The imbalanced data had the same effect on random forest as it had on the classification tree. The random forest had no predictive value with the data set and therefore undersampling was utilized. Oversampling was not applicable as the data set became too large. R and the computer used in the research could not run the oversampled data.

Undersampling was conducted with the training set and sampling was set to 29 412 which is two times the number of defaulted loans in “TARGET” variable.

5.4.3.2 Prediction

For the RandomForest function, the number of trees was set to 500 and the number of variables per level (mtry) to three. Number of trees indicate the number of trees in the model, whereas mtry displays how many randomly sampled variables are used at each split. Mtry was chosen by using a random search with the caret package. The data used in the RandomForest function was the undersampled and cross-validated data which was created with the training set. For prediction function test set was used as “newdata”.

5.4.4 Extreme Gradient Boosting (XGBoost)

For Extreme Gradient Boosting (XGBoost) the split was executed similarly to the previous models. XGBoost is different from the previous models as it only handles numeric vectors. Numeric values and character values were left to their original form and not transformed into factors. The next section introduces one-hot encoding that was executed on the character variables.

5.4.4.1 One-hot Encoding

As mentioned, the data set had numeric values and character variables. One-hot encoding was used to turn character variables into numeric vectors for them to be applicable to XGBoost. One-hot encoding was executed with the spars matrix and the spars.model.matrix function. Using the function allowed to create a spars matrix that would be used as an input to the model and simultaneously one-hot encode the character values. All character values, except the “TARGET” variable, were transformed into numeric and binary values. This was executed both on the training set and the test set.

The “TARGET” variable from the training set and the test set was saved to new objects called “train_label” and “test_label”. The “TARGET” variable was saved separately to ensure its values are not transformed. They were used later in training the model and evaluating performance.

5.4.4.2 Model Creation

The XGBoost hyperparameters were tuned with the training set for better performance. The default booster for XGBoost, gbtree, was used in the model and the "object" was set to "binary:logistic" as "TARGET" variable is binary. After trying with different parameters, the tree depth was set to five, eta to 0.03 and gamma to one. Eta controls the rates at which the model learns patterns in the data. Gamma in XGBoost is used to reduce overfitting. Gamma's default value is 0, which would not penalize coefficients that do not improve performance. These parameters gave the best results on the AUC and in the confusion matrix. In addition, "nrounds" was set to 500. "Nrounds" in XGBoost is similar to number of trees in the random forest model. The hyperparameters were saved in order to place them into model training.

XGBoost function was used in model training and the hyperparameters were set to the function. The data used in the function was the training data that was transformed into a sparce matrix before. As "label" in the function was "train_label" which contained the values from the "TARGET" variable.

5.4.4.3 Prediction

Prediction for XGBoost was executed with the predict function. The test set was used as "newdata" for the prediction. In addition, the training set that was trained with the XGBoost model was used in the predict function as "object". Predictions created were saved to new elements in order to evaluate the model with the confusion matrix, the ROC curve and to find the optimal threshold value with Youden's index.

5.5 Results

This section displays the prediction results achieved from logistic regression, classification tree, random forest and XGBoost. Predictions were performed on the test set on every model in order to compare each model's prediction power. As mentioned in section 5.4, the k-fold cross-validation was utilized to ensure proper evaluation on unseen data and to reduce bias. The ROC curve and confusion matrix were used to evaluate the prediction power. As the predictions and their power with unseen data is of interest, only the test set results are displayed in this section. The AUC shows how much predictive value the model has on a scale from 0.5 to 1. The aim is to obtain a value close to one as

a value of 0.5 means that the prediction is a coin toss and value of 1 means that the model's predictions are 100% right.

5.5.1 Logistic Regression

5.5.1.1 ROC Curve and the AUC

After the model was trained with the training set and the prediction was executed with the test set, the ROC curve was drawn. The ROC curve was created with the roc.curve function. The test set's "TARGET" column was set as the response objective to the function. The prediction that was created with the test set and the trained logistic regression model was set as the predicted value. Figure 29 displays the ROC curve and the AUC for logistic regression. The ROC curve displays an AUC of 0.6754 for the test set. After creating the ROC curve and the AUC, Youden's index was applied to find the threshold that maximizes sensitivity and specificity. The Youden's index threshold was used in creating the confusion matrix, which is introduced next.

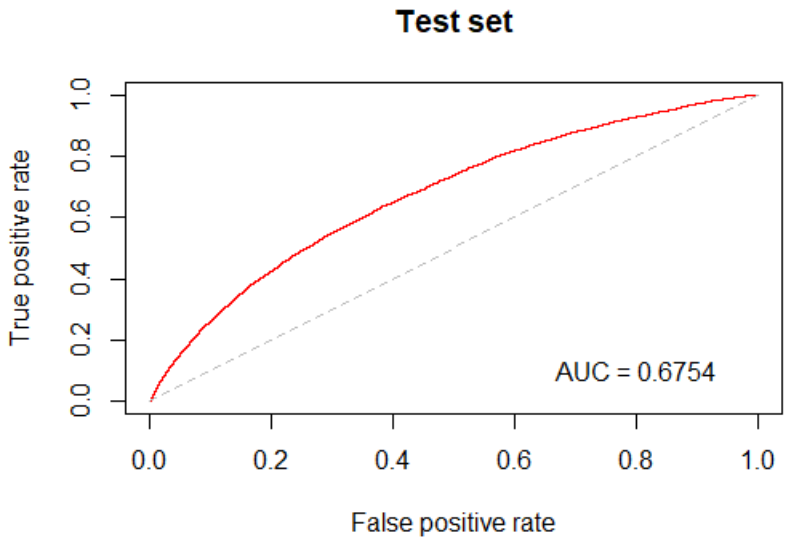

Figure 29. *ROC Curve and the AUC for the logistic regression's set test set*

5.5.1.2 Confusion Matrix

The purpose of the confusion matrix in the research was to obtain information on sensitivity and specificity. For loan defaults, both are important as banks do not want to reject qualified customers nor they want to give loans to unqualified customers as their default could lead to losing the profits made from qualified customers.

The logistic regression model predicted 63.3% of the defaulted loans correctly and 62.1% of the non-defaulted loans. Accuracy is also presented in table 1, although it is not the most reliable metric for this research. For logistic regression, the threshold value that maximized specificity and sensitivity was 0.0903 for the test set.

Logistic regression	Accuracy	AUC	Sensitivity	Specificity	Threshold
Test set	62.1%	0.675	63.3%	62.1%	0.0903

Table 1. *Confusion matrix results from the logistic regression model*

5.5.2 Classification Tree

Creating the ROC curve and the confusion matrix for the classification tree was executed the same way as in logistic regression. As oversampling and undersampling were used in the model creation, this section presents the results on oversampled and undersampled test sets.

5.5.2.1 ROC Curve and the AUC

Figure 30 displays the ROC curve for the oversampled test set. As mentioned previously in the model creation section 5.4.2.1, the classification tree had no predictive power before oversampling or undersampling as the AUC was 0.5. With oversampling the predictive value increased and the AUC was 0.6246.

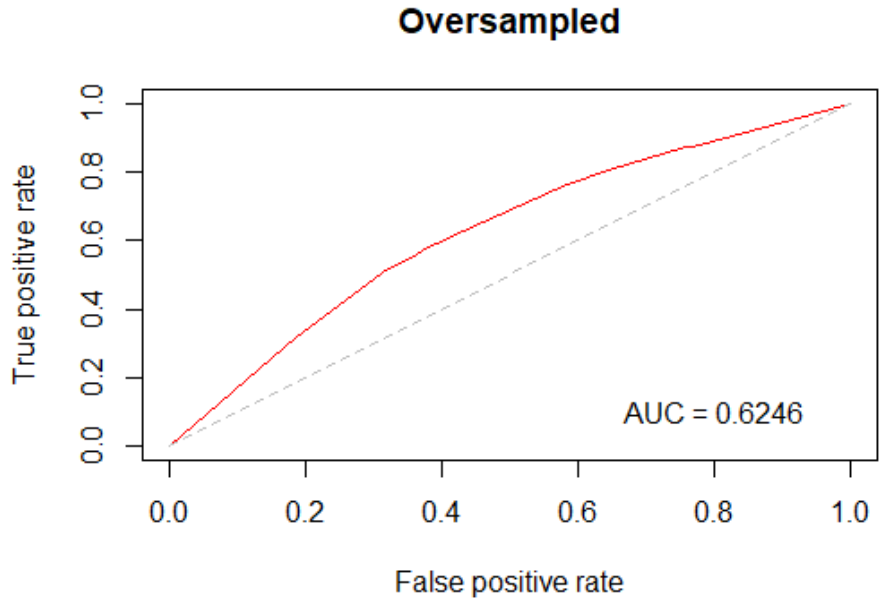

Figure 30. *The ROC curve and the AUC for classification tree's oversampled test set*

The results on undersampled data are displayed in figure 31. The AUC increased slightly to 0.6296 compared to the oversampled test set. As oversampling, undersampling was able to increase the predictive power compared to the data set where oversampling or undersampling was not executed.

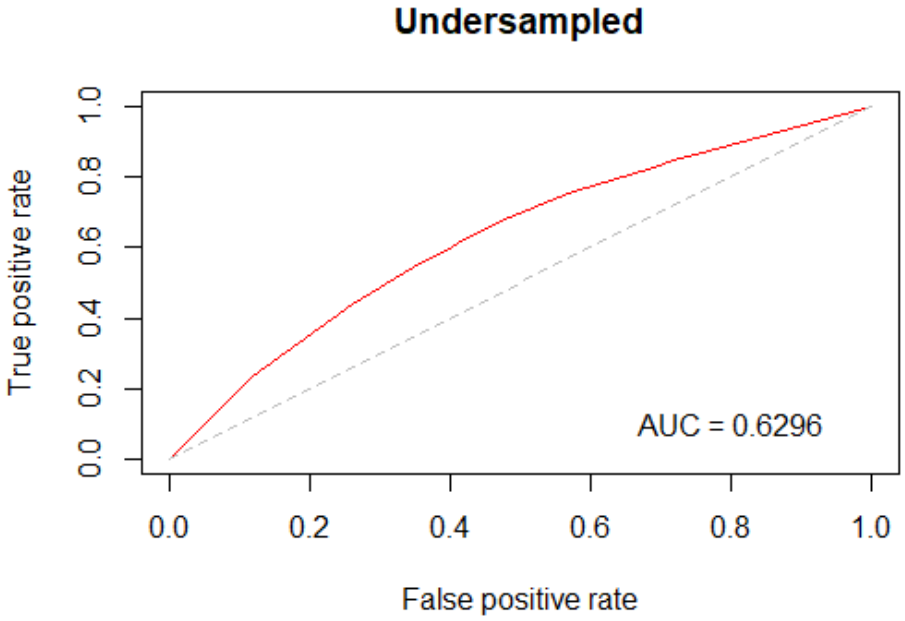

Figure 31. *The ROC curve and the AUC for classification tree's undersampled test set*

5.5.2.2 Confusion Matrix

The AUC's in figure 30 and 31 show that the prediction power in the oversampled and undersampled data is almost identical. Oversampled data performed better in predicting specificity (non-default) and the undersampled in predicting sensitivity (default), which can be seen in table 2 and 3, respectively. The confusion matrix in table 2 presents that the oversampled data predicts specificity better than sensitivity. In the research it means that non-defaulted loans are predicted better than defaulted loans. As stated before, predicting the defaulted loans correctly is more important in loan granting. Based on the sensitivity and AUC, the undersampled data performs slightly better than the oversampled data and it will be used when comparing the results to other models.

Classification tree	Accuracy	AUC	Sensitivity	Specificity	Threshold
Oversampled test set	61.8%	0.625	57.9%	62.2%	0.512

Table 2. *Confusion matrix results from classification tree's oversampled model*

Classification tree	Accuracy	AUC	Sensitivity	Specificity	Threshold
Undersampled test set	53.2%	0.630	68.7%	51.6%	0.458

Table 3. *Confusion matrix results from classification tree's undersampled test set*

5.5.3 Random Forest

For random forest, the ROC curve and the confusion matrix were executed the same way as for the previous models. As mentioned in section 5.4.3.1, undersampling was executed on training set but oversampling was not possible as the data set created became too large to run for R and the computer used in the research.

5.5.3.1 ROC Curve and AUC

Figure 32 displays the ROC curve and the AUC for the random forest model. Without undersampling, the model had no predictive power as the AUC was 0.5. With undersampling, AUC rose to 0.677, which is highest of the models so far.

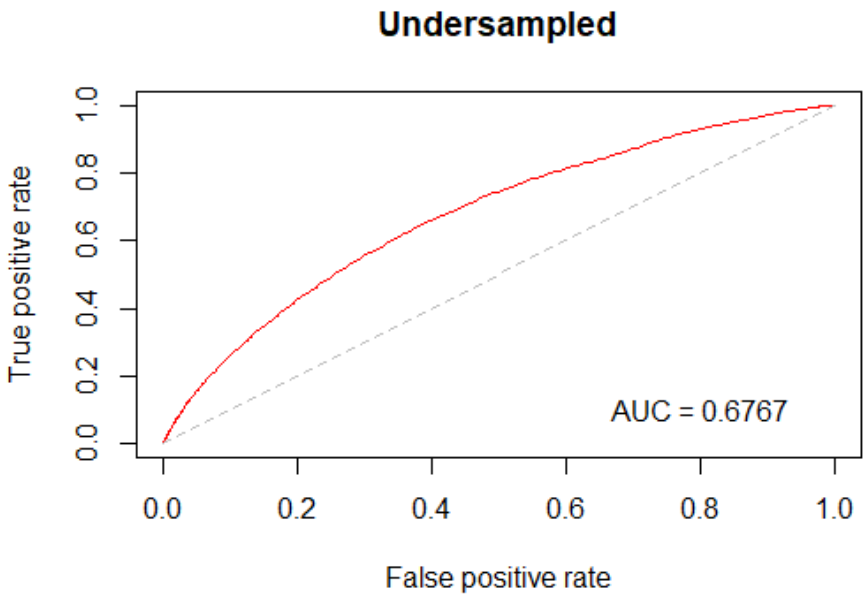

Figure 32. *Roc curve for the random forest*

5.5.3.2 Confusion Matrix

The confusion matrix in table 4, shows that the random forest model predicts non-defaulted loans better than defaulted loans. Overall, the results are the best ones among the models used so far. In addition to AUC, specificity is highest of the models.

Random forest	Accuracy	AUC	Sensitivity	Specificity	Threshold
Undersampled test set	64.1%	0.677	62.1%	64.2%	0.499

Table 4. *Confusion matrix results from random forest*

5.5.4 Extreme Gradient Boosting (XGBoost)

The performance metrics for evaluating prediction power for XGBoost were the ROC curve and confusion matrix with the threshold value obtained from the Youden's index. The results on the test set are presented below.

5.5.4.1 ROC Curve and the AUC

The ROC curve and the AUC for XGBoost are displayed in figure 33. The test set confirmed XGBoost's predictive power compared to other models as the AUC was 0.695.

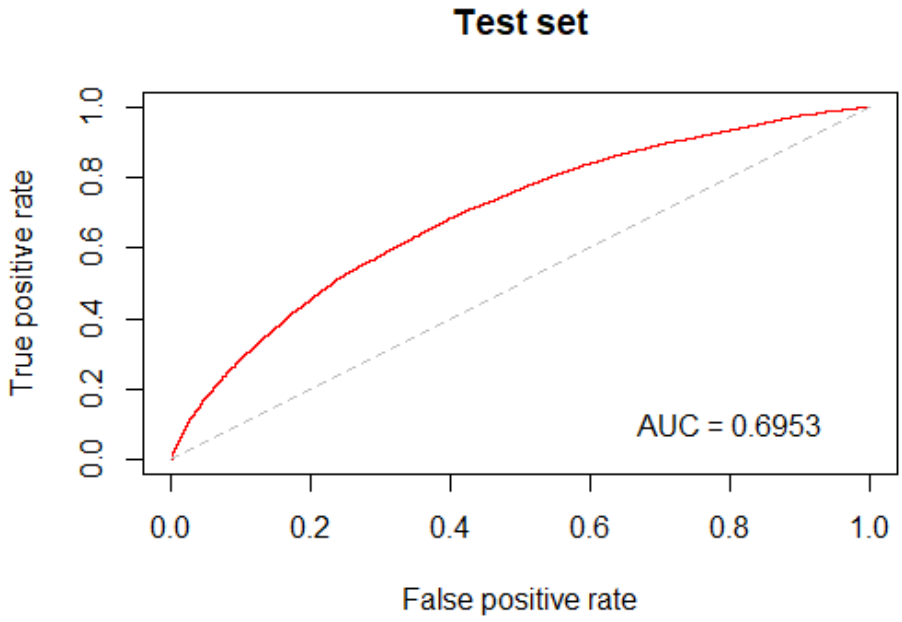

Figure 33. *ROC curve and AUC for XGBoost*

5.5.4.2 Confusion Matrix

The confusion matrix in table 5 enhances what the ROC curve displayed. The sensitivity on the test set is 0.700, highest of all the models used in the research. The XGBoost identifies 70% of defaulted loans correctly on unseen data. Specificity in XGBoost is 0.585 for the test set. As mentioned, Youden's index was used in the confusion matrix for setting the optimal threshold value for the model. The threshold value that maximizes sensitivity and specificity was 0.462 for the test set.

XGBoost	Accuracy	AUC	Sensitivity	Specificity	Threshold
Test set	59.5%	0.695	70.0%	58.5%	0.462

Table 5. *Confusion matrix results from XGBoost*

5.5.5 Results Analysis

The results from all models are displayed in table 6 for proper comparison. All the models were trained with the training set, but the results are presented based on the predictions made with the test set. Accuracy is not an essential metric in the research as the data is heavily imbalanced between non-defaulted loans and defaulted loans as presented in section 5.2.

In addition to accuracy, table 6 presents the AUC from the ROC curve. Sensitivity displays how well the model predicts defaulted loans and specificity shows the prediction on non-defaulted loans. Finally, the threshold is the Youden's index threshold that was used in the confusion matrix.

	Accuracy	AUC	Sensitivity	Specificity	Threshold
Logistic regression	62.1%	0.675	63.3%	62.1%	0.0903
Classification tree (Undersampled)	53.2%	0.630	68.7%	51.6%	0.458
Random forest	64.1%	0.677	62.1%	64.2%	0.499
XGBoost	59.5%	0.695	70.0%	58.5%	0.462

Table 6. *Results from used models*

For the classification tree, the results in section 5.5.2.1 show that the undersampled and oversampled data performed close to equal in predicting power according to AUC. The

undersampled model was chosen for results analysis as the AUC and sensitivity were higher than in the oversampled model. As mentioned in section 5.5.1.2, sensitivity in this research is important. The objective was to predict both defaulted loans and non-defaulted loans precisely.

The classification tree's undersampled model predicted sensitivity with 68.7% precision. Sensitivity was coherent with the sensitivity of logistic regression and random forest. The other evaluation metrics were far from optimal in classification tree, even if sensitivity was close to XGBoost's default prediction.

The AUC indicates that the prediction power in classification tree was not as powerful as in other models. As presented in figure 34, classification tree had the lowest ROC curve (AUC 0.630), while the second lowest was logistic regression's 0.675. Random forest had an AUC of 0.677 and the other evaluation metrics were close to logistic regression as well. XGBoost's AUC of 0.695 was the highest of implemented models.

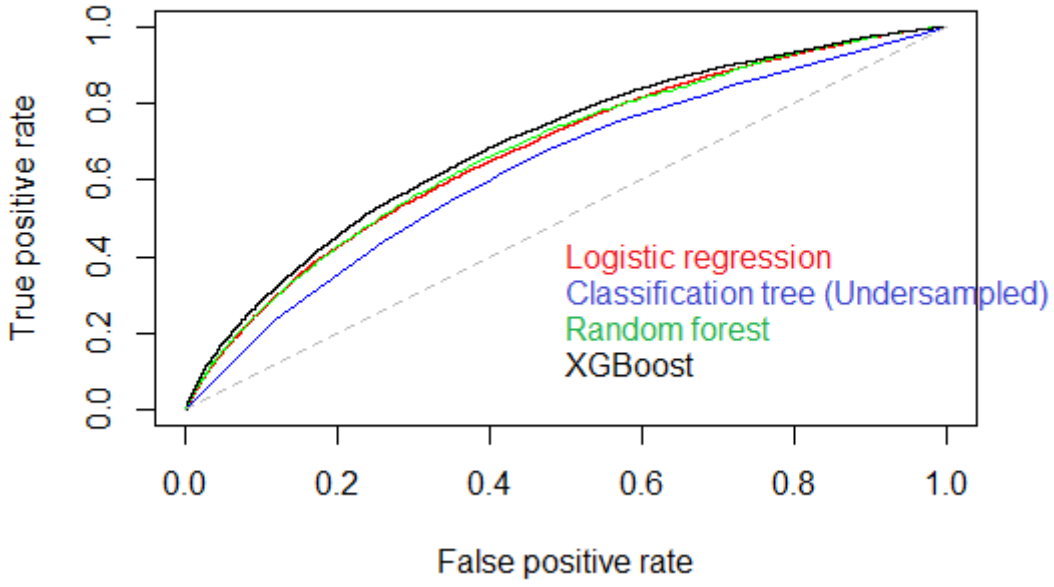

Figure 34. *Comparison of ROC curves*

Logistic regression is not a tree-based model as the other models and was chosen for the research to have a benchmark. Still, the result on logistic regression outperformed classification tree in AUC and specificity and random forest in sensitivity. However, the

random forest model outperformed other models in predicting non-defaulted loans as specificity is 64.2%.

Based on these results, the XGBoost model obtained the most precise and powerful results on loan default prediction based on the AUC and sensitivity. As the most powerful model was recognised, the most important variables in making the prediction with XGBoost were identified. They will be introduced when answering research questions in the next chapter.

6 DISCUSSION

Previous research on loan defaults have compared different learning algorithms based on evaluation metrics similar to this research. This study was conducted with the literature review and the empirical study where different algorithms were compared. Literature review explained how machine learning is utilized in loan granting. In addition, it explained different types of machine learning, learning algorithms and how data is prepared, and models built for the predictions. The empirical study was executed with four machine learning models and the aim was to identify the most powerful model. The models used in the study were logistic regression, classification tree, random forest and XGBoost, which were compared based on chosen evaluation metrics sensitivity, specificity and the area under the ROC curve.

Logistic regression and classification tree were chosen as they are commonly used classification methods in binary classification problems and offer a benchmark for predictions. In addition, as decision trees are the base of random forest models, it was of interest to compare classical models to more advanced models. In this chapter, first the results on logistic regression and classification tree are discussed. Second, results from more advanced models, random forest and XGBoost, are compared followed by answers to research questions.

Logistic regression model achieved an AUC of 0.675, a sensitivity of 63.3% and a specificity of 62.1% as presented in table 7. Classification tree's AUC was 0.630, sensitivity 68.7% and specificity 51.6%. Logistic regression outperformed classification tree in predicting non-defaulted loans and in prediction power based on the AUC. Logistic regression is a widely used binary classification method and several studies have utilized it in loan default predictions. As mentioned above, logistic regression was chosen to have a benchmark model and to compare it to more advanced models. The results achieved with logistic regression are close to random forest model, which had an AUC of 0.675, a sensitivity of 62.1% and a specificity of 64.2%. The results display why logistic regression is widely used in binary classification problems. Classification tree's overall results underperformed compared to other models. Predicting defaulted loans was second highest, however overall results and prediction power was the lowest of the models.

	Accuracy	AUC	Sensitivity	Specificity	Threshold
Logistic regression	62.1%	0.675	63.3%	62.1%	0.0903
Classification tree (Undersampled)	53.2%	0.630	68.7%	51.6%	0.458

Table 7. *Results from logistic regression and classification tree*

To conclude the discussion on the benchmark models logistic regression and classification tree, the models had moderate results compared to other studies. The models were able to classify majority of the qualified and unqualified applicants correctly as can be seen in sensitivities and specificities. Comparison between studies is complicated as data sets are different. However, compared to results by Silva et al. (2020) our model is more applicable to real-life scenarios. Their study predicted only non-defaults promptly as their logistic regression model had a sensitivity of 0.94% and a specificity of 99.55% when applied to all data available. In real-life scenarios, financial institutes using their model would have an enormous number of defaulting customers.

As mentioned above, random forest and XGBoost were chosen as they are more advanced models. The aim was to have a mixture of benchmark models and more advanced models for comparison. In addition, as XGBoost has been dominating many machine learning competitions, it was of interest to see how well it performs on an imbalanced data set compared to more traditional methods. Table 8 presents the results on random forest and XGBoost.

	Accuracy	AUC	Sensitivity	Specificity	Threshold
Random forest	64.1%	0.677	62.1%	64.2%	0.499
XGBoost	59.5%	0.695	70.0%	58.5%	0.462

Table 8. *Results from random forest and XGBoost*

As mentioned above, the results on random forest were slightly better than in logistic regression. From the two more advanced models, XGBoost outperformed random forest in every metric except specificity. XGBoost's AUC outperformed other models distinctly and it was the only model reaching 70% in sensitivity. However, specificity on XGBoost was only the third highest while random forest outperformed other models in specificity. XGBoost outperforming other models in overall result was no surprise. As mentioned, XGBoost has been widely used in Kaggle competitions. In addition, Xia et al. (2017) compared logistic regression, random forest and XGBoost in their study and XGBoost achieved the highest results.

Furthermore, the following research questions were proposed in the introduction chapter:

1. *How is loan granting regulated?*

2. *How is machine learning utilized to facilitate decision making in loan granting?*

3. *Which of the selected machine learning models is the most effective in loan default prediction?*

4. *Which variables are the most important in predicting a loan default?*

The first research question asks how loan granting is regulated and it can be answered based on chapter 2. As mentioned, the regulations banks face are involuntary and violating them could result in immense fines. The Basel Agreement is a regulation, which affects banks and is evolved continuously to secure their resiliency. In addition, governments order laws and regulations to ensure that credits are not granted for unqualified customers, to mitigate over-indebtedness and to secure that the housing market remains stable. Loan-to-value ratio and loan cap are examples of such regulations in the Finnish society and they have proven to be effective in controlling the functionality of housing markets. In addition, the forthcoming implementation of positive credit register underlines how agile banks must be in the changing regulatory environment.

The second research question can be answered based on the literature review. Granting loans is core business in banking. Digitalization, competition due to low interest rates and the ease of applying for loans, have increased the number of loan applications tremendously. For a bank to be competitive, it has to be effective in processing loan applications. If they can minimize the time spent with applications that will be declined, they can spend the time on loans that will be granted. Without machine learning, the procedure is impossible as every application must be processed manually. As banks cannot become too careful in loan granting, machine learning is utilized together with manual labour for efficient loan application process. In addition, machine learning enables following regulations. As explained in section 1.1.2, the Basel Agreement obliges banks to calculate credit risk components, such as default probability, which is possible with machine learning algorithms.

The third research question concerns which is the most effective machine learning model in loan default prediction. Based on the study, the most effective model is XGBoost, which was also expected to be the most effective. The XGBoost was able to predict between the defaults and non-defaults with a 69.5% chance based on the AUC. In addition, it was able to predict the defaulted loans correctly 7 times out of 10, thus being the only model reaching 70% in sensitivity.

The final question concerning the most important variables in loan default prediction is answered in figure 35. The figure displays the five most important variables in the most effective model, XGBoost. The most important variables were extracted from the XGBoost model with the xgb.importance function. The function shows variable importance based on average gain. Gain indicates how a feature makes decision tree's branch purer. (RDocumentation, n.d.).

	Feature	rank
1:	YearsWorked	1
2:	LTV	2
3:	Years_since_Phone_Change	3
4:	AMT_GOODS_PRICE	4
5:	AMT_ANNUITY	5

Figure 35. *Most important variables in creating XGBoost model*

"YearsWorked", consisting of employment years, was the most important variable in creating the XGBoost model. "YearsWorked" being most important is not surprising as longer employment tends to result in more savings. In addition, years employed usually relates to stability in workplace. Savings enable making loan payments even when sudden changes, such as becoming ill and staying on a sick leave, occur. Dramatic changes usually contribute to financial problems, which contribute to loan defaults.

The second important variable is "LTV" (Loan-to-value ratio). As mentioned in the introduction, "LTV" is a new regulation in Finnish mortgages. Loan-to-value ratio is used to secure that the housing market functions properly. The Covid-19 crisis exemplified that LTV is a powerful tool to secure that the market functions properly as the rate can be adjusted. According to this study, it is also a relevant regulation to prevent loan defaults.

It is impossible to explain why "Years_since_Phone_Change" is the third most important variable. The fourth variable, "AMT_GOODS_PRICE", indicates how much a good that was purchased with the granted loan cost. As a conclusion, it could mean that the more expensive purchases, such as a house lead to more loan defaults than smaller purchases. The fifth variable "AMT_ANNUITY" describes how much loan annuity is. The annuity being in the top 5 is not a surprise. The larger the annuity is on the loan, the more challenges are faced with loan payments due to sudden changes.

To conclude the discussion, in the research more advanced models, XGBoost and random forest, outperformed benchmark models logistic regression and classification tree. However, the predictive power and ability to distinguish between defaulted and non-defaulted loans of logistic regression compared to random forest was close and surprising. The results on sensitivity and specificity are coherent to other studies presented in section 3.1. However, the results on AUC's could have been higher. The best AUC score was achieved with the XGBoost model, which had an AUC of 0.695. The perfect model would have had an AUC of 1 and the worst 0.5. As the threshold was varied to maximize sensitivity and specificity in our study, the majority of defaulting customers and customers with no defaults were predicted correctly.

The results underline the challenge with real-life problems: Loan defaults are minority classes in every country and data imbalance complicates the predictions. Fortunately,

defaults are minorities as it keeps the financial sector stable. The limitations of the research and recommendations for future studies are presented in the next section.

6.1 Limitations and Future Research

The conducted study has limitations, which affect the study results. The data set consisted of three different Excel files. One data set contained information about the latest application, the second had information on previous applications and the third contained column description. Only the data set with information on the latest application was used. According to the researcher, the information on previous applications was irrelevant.

The data set used in the research had many important variables, which are used in Finnish banks' loan applications. In turn, many of the variables had to be removed as descriptions were unclear or missing. Also, it was mentioned that the data set is a real-life data set but the origin or the country was not mentioned. However, the data set did not include all important variables. A very interesting variable would have been credit score. Credit score could be a number between 0 to 5, meaning that if an applicant has a credit score of 0, he or she has had problems with previous loans or account overdrafts. If the score is 5, the applicant has paid every payment on due and has maintained accounts accordingly.

Another limitation is the computing power of the computer the research was executed on. Oversampling the data set for random forest made the data too large and it could not be executed. The problem also rose when the second data set was tried to utilize for more predictive power.

For future research, the loan default prediction should be executed for a mortgage data set. The rules and restrictions on mortgages are different and stricter than for revolving loans. Revolving loans do not need collaterals in the Finnish society, so the loan-to-value ratio does not apply to them. Loan-to-value ratio is calculated based on the price of the house bought or other collaterals. A mortgage data set would enable studying the significance of the new restriction further. In addition, section 5.3.2 introduced the issue of values "365243" in the column "DAYS_EMPLOYED", which could be addressed in future research. The values were deleted from the data set in this research for simplicity. However, deleting the values led to deleting 51 232 observations, which was 71.7% of all deleted observations. Future research should solve how to maintain these values.

REFERENCES

Abbott, D. (2014). *Applied predictive analytics: Principles and techniques for the professional data analyst.* John Wiley & Sons, Incorporated

Abid, L., Masmoudi, A. & Zouari-Ghorbel, S. (2018). The Consumer Loan's Payment Default Predictive Model: an Application of the Logistic Regression and the Discriminant Analysis in a Tunisian Commercial Bank. *Journal of the Knowledge Economy, 9*(3), 948-962. https://doi.org/10.1007/s13132-016-0382-8

Alpaydin, E. (2014). *Introduction to machine learning* (3rd ed.). MIT Press

Amadeo, K. (2020, September 17). *The Causes of the Subprime Mortgage Crisis*. The Balance. Retrieved from https://www.thebalance.com/what-caused-the-subprime-mortgage-crisis-3305696

Anderson, D.R., Sweeney, D.J. & Williams, T.A. (2010). *Statistics for Business and Economics.* (11th ed.). Cengage Learning, Inc.

Awad, M. & Khanna, R. (2015). *Efficient learning machines: Theories, concepts, and applications for engineers and system designers.* Apress. https://doi.org/10.1007/978-1-4302-5990-9

Bandyopadhyay, A. (2016). *Managing portfolio credit risk in banks.* Cambridge University Press. https://doi.org/10.1017/CBO9781316550915

Bank of Finland (2019, September 23). *European Systemic Risk Board recommends new measures to Finland to address household indebtedness*. Contify Banking News. https://link.gale.com/apps/doc/A600532602/STND?u=aboacad&sid=STND&xid=1e8c7219

Bell, J. (2014). *Machine Learning: Hands-On for Developers and Technical Professionals.* John Wiley & Sons, Incorporated

Bessis, J. (2015). *Risk management in banking (4th ed.)*. John Wiley & Sons, Incorporated

Blöchlinger, A & Leippold, M. (2006). Economic benefit of powerful credit scoring. *Journal of Banking and Finance, 30*(3), 851-873. Retrieved from https://doi.org/10.1016/j.jbankfin.2005.07.014

Brown, I. & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications, 39*(3), 3446-3453. https://doi.org/10.1016/j.eswa.2011.09.033

Brown, M. S. (2014). *Data Mining for Dummies.* John Wiley & Sons Incorporated

Brownlee, J. (2021, January 5*). Random Oversampling and Undersampling for Imbalanced Classification*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

Brownlee, J. (2019, December 23). *A Gentle Introduction to Imbalanced Classification*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/what-is-imbalanced-classification/

Brownlee, J. (2016a, August 17) *A Gentle Introduction to XGBoost for Applied Machine Learning*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

Brownlee, J. (2016b, February 5) *Tune Machine Learning Algorithms in R (random forest case study*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/

Bullivant, G. (Ed.). (2016). *Credit management (6th ed.).* Taylor & Francis Group

Chakure, A. (2019, June 29th). *Random Forest Regression: Along with its implementation in Python.* Medium. Retrieved from https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f

Chang, Y.C., Chang, K.H., Chu, H.H. & Tong, L.I. (2016). Establishing decision tree-based short-term default credit risk assessment models. *Communications in Statistics – Theory and Methods*, *45*(23), 6803-6815. https://doi.org/10.1080/03610926.2014.968730

Chen, T. & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785-794. https://doi.org/10.1145/2939672.2939785

European Central Bank (2004). *Financial stability review.* Retrieved from https://www.ecb.europa.eu/pub/financial-stability/fsr/html/all_releases.en.html

Financial Supervisory Authority (2015). *Regulations and guidelines on the calculation of loan-to-value ratio enter into force on 1 July 2016.* Retrieved from https://www.finanssivalvonta.fi/en/publications-and-press-releases/supervision-releases/2015/regulations-and-guidelines-on-the-calculation-of-loan-to-value-ratio-enter-into-force-on-1-july-2016/

Financial Supervisory Authority (2018, December 2), *Housing loans and loan cap.* Retrieved from https://www.finanssivalvonta.fi/en/Consumer-protection/banking-services/housing-loans-and-loan-cap/

Financial Supervisory Authority (2021, June 29), *Macroprudential decision: Housing loan cap for residential mortgage loans other than first-home loans to be set at 85%.* Retrieved from https://www.finanssivalvonta.fi/en/publications-and-press-releases/Press-release/2021/macroprudential-decision-housing-loan-cap-for-residential-mortgage-loans-other-than-first-home-loans-to-be-set-at-85/

Finnish Competition and Consumer Authority (2020). *Maximum interest rate and marketing of credits will be limiter temporarily due to corona.* Retrieved from https://www.kkv.fi/en/current-issues/press-releases/2020/1.7.2020-maximum-interest-rate-and-marketing-of-consumer-credits-will-be-limited-temporarily-due-to-corona/

Galar, M., Fernández, A., Barrenechea, E., Bustince, H. & Herrera, F. (2012). A review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463-484. https://www.doi.org/10.1109/TSMCC.2011.2161285

Gandhi, R. (2018a.) *Boosting Algorithms: AdaBoost, Gradient Boosting and XGBoost*. Medium. Retrieved from https://medium.com/hackernoon/boosting-algorithms-adaboost-gradient-boosting-and-xgboost-f74991cad38c

Gandhi, R. (2018b.) *Gradient Boosting and XGBoost*. Medium. Retrieved from https://medium.com/hackernoon/gradient-boosting-and-xgboost-90862daa6c77

Gonzalez, A. (2017, September 6). *ELI5: ROC curve, AUC metrics*. Medium. Retrieved from https://medium.com/@andygon/eli5-roc-curve-auc-metrics-ac4fe482f018

Hackerearth, (n.d.). *Beginners Tutorial on XGBoost and Parameter Tuning in R*. Retrieved from https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/

Helenius, T. (2018). *Miksi korot ovat niin matalat?* Taloustaito. Retrieved from https://www.taloustaito.fi/Blogi/blogit-2018/miksi-korot-ovat-niin-matalat/#719cb4f9

Hosmer, D. W. J., Lemeshow, S. & Sturdivant, R. X. (2013). *Applied logistic regression (3rd ed)*. John Wiley & Sons, Incorporated

Lee, T.S., Chiu, C.C., Chou, Y.C. & Lu, C.J. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics & Data Analysis. 50*(4), 1113-1130. https://doi.org/10.1016/j.csda.2004.11.006

Lessmann, S., Baesens, B., Seow, H. & Thomas, L.C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research, 247*(1), 124-136. https://doi.org/10.1016/j.ejor.2015.05.030

Ince, H & Aktan, B. (2010). A comparison of data mining techniques for credit scoring in banking: A managerial perspective. *Journal of Business Economics and Management*, 10(3), 233-240. https://doi.org/10.3846/1611-1699.2009.10.233-240

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer. Retrieved from https://link.springer.com/book/10.1007/978-1-4614-7138-7#toc

Kabacoff, R.I. (2015). *R in Action: Data analysis and graphics with R (2nd ed)*. Manning Publications Co.

Kamath, R.S., & Kamat, R.K. (2016). *Educational data mining with R and Rattle.* River Publishers

Kotu, V. & Deshpande, B. (2014). *Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner.* Elsevier Science & Technology

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence, 5*(4), 221-232. https://doi.org/10.1007/s13748-016-0094-0

Krzanowski, W.J. & Hand, D.J. (2009). *ROC curves for continuous data.* Chapman and Hall/CRC. https://doi.org/10.1201/9781439800225

Larose, D.T. & Larose, C.D. (2015). *Data mining and predictive analytics.* John Wiley & Sons, Incorporated

Magnus, M., Margerit, A., Mesnard, B. & Korpas, A. (2017). *Upgrading the Basel standards: from Basel III to Basel IV?* European parliament – Economic governance support unit. Retrieved from https://www.europarl.europa.eu/thinktank/en/document.html?reference=IPOL_BRI(2016)587361

Ministry of Justice (2021, March 30.) *Positiivisella luottorekisterillä halutaan torjua ylivelkaantumista.* Ministry Of Justice. Retrieved from https://oikeusministerio.fi/en/-/positiivisella-luottotietorekisterilla-halutaan-torjua-ylivelkaantumista

Mithrakumar, M. (2019). *How to tune a Decision Tree?* Towards Data Science. Retrieved from https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680

Mueller, J.P. & Massaron, L. (2016). *Machine learning for dummies*. John Wiley & Sons, Incorporated

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective.* MIT Press

Nickolas, S. (2021, May 21). *What do correlation coefficients positive, negative, and zero mean?* Investopedia. Retrieved from https://www.investopedia.com/ask/answers/032515/what-does-it-mean-if-correlation-coefficient-positive-negative-or-zero.asp

Nordea (2021). *Information about the cap on and the self-financing share of housing loans.* Retrieved from https://www.nordea.fi/en/personal/our-services/loans/home-loans/topical-information-about-the-cap-on-housing-loans.html

Odegua, R. (2020). *Predicting bank loan default with extreme gradient boosting.* arXiv.org. Retrieved from https://arxiv.org/ftp/arxiv/papers/2002/2002.02011.pdf

Pykes, K. (2020, September 10). *Oversampling and Undersampling: A Technique for Imbalanced Classification*. Towards Data Science. Retrieved from https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf

Raijas, A., Lehtinen, A.R. & Leskinen, J. (2010.) Over-Indebtedness in the Finnish Consumer Society. *Journal of Consumer Policy 33*(3), 209-223. https://doi.org/10.1007/s10603-010-9131-8

Ramzai, J. (2020). *Why is model validation so darn important and how is it different from model monitoring.* Towards Data Science. Retrieved from https://towardsdatascience.com/why-is-model-validation-so-darn-important-and-how-is-it-different-from-model-monitoring-61dc7565b0c

RDocumentation, (n.d.). *xgb.importance: Show importance of features in a model*. RDocumentation.org, Retrieved from https://www.rdocumentation.org/packages/xgboost/versions/0.6.4.1/topics/xgb.importance

Shmueli, G. & Koppius, O.R. (2011). Predictive Analytics in Information Systems Research. *MIS quarterly, 35*(3), 553-572. https://doi.org/10.2307/23042796

Silva, E.C., Lopes, I.C., Correia, A. & Faria, S. (2020). A logistic regression model for consumer default risk. *Journal of Applied Statistics, 47*(13-15), 2879-2894. https://doi.org/10.1080/02664763.2020.1759030

Silver, C. (2021). *Over 10 years later, lessons from the 2008 financial crisis.* Investopedia. Retrieved from https://www.investopedia.com/news/10-years-later-lessons-financial-crisis/

Statistics Finland (2021, January 28). *Debts of household-dwelling units with housing loans are more than double their annual income.* Statistics Finland. Retrieved from https://www.stat.fi/til/velk/2019/velk_2019_2021-01-28_tie_002_en.html

Statistics Finland (2013, June 6). *Talouskriisi on kohdellut lempeästi suomalaisten asumista*. Statistics Finland. Retrieved from https://www.stat.fi/artikkelit/2013/art_2013-03-11_008.html?s=0

Tian, Z., Xiao, J., Feng, H. & Wei, Y. (2020). Credit Risk Assessment based on Gradient Boosting Decision Tree. *Procedia Computer Science, 174,* 150-160. https://doi.org/10.1016/j.procs.2020.06.070

Uusitalo, K. (2019). *Pikavippimarkkinat muuttuvat nyt, ja tässä ovat seuraukset: lainansaanti vaikeutuu, maksuhäiriöt lisääntyvät – monen velkakierre voi myös katketa.* Yle. Retrieved from https://yle.fi/uutiset/3-10943120

Xia, Y., Liu, C. & Liu, N. (2017). Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending. *Electronic Commerce Research and Applications,* 24, 30-49. https://doi.org/10.1016/j.elerap.2017.06.004

Xu, J., Lu, Z. & Xie, Y. (2021). Loan default prediction of Chinese P2P market: a machine learning methodology. *Scientific Reports,* 11. https://doi.org/10.1038/s41598-021-98361-6

Yang, J., Chen, Y., Zhang, C., Park, D.S. & Yoon, S. (2018). *Introductory chapter: Machine Learning and Biometrics.* IntechOpen. Retrieved from https://www.intechopen.com/chapters/62526

Yin, J. & Tian, L. (2014). Join inference about sensitivity and specificity at the optimal cut-off point associated with Youden index. *Computational Statistics & Data Analysis, 77,* 1-13. https://doi.org/10.1016/j.csda.2014.01.021

Youden, W.J. (1950). Index for rating diagnostic tests. *Camcer.* 3, 32-35 https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3

Yu, Y. (2020, August). The Application of Machine Learning Algorithms in Credit Card Default Prediction. *2020 International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA. https://doi.org/10.1109/CDS49703.2020.00050

Wang, Z., Jiang, C., Ding, Y., Lyu, X. & Liu, Y. (2018). A Novel behavioral scoring model for estimating probability of default over time in peer-to-peer lending. *Electronic Commerce Research and Applications,* 27, 74-82. https://doi.org/10.1016/j.elerap.2017.12.006

Zhu, L., Qiu, D., Ergu, D., Ying, C. & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, 162, 503-513. https://doi.org/10.1016/j.procs.2019.12.017