The Power of Credit Scoring: Evaluating Machine Learning and Traditional Models in Swedish Retail Banking

UNIVERSITY OF GOTHENBURG
SCHOOL OF BUSINESS, ECONOMICS AND LAW

Written by:
Emma von der Burg
Saga Strömberg

Supervisor: Marcin Zamojski
Master's thesis in Finance, 30 hec
Spring 2023
Graduate School, School of Business, Economics and Law, University of Gothenburg

Abstract

In this paper, we investigate and compare different credit scoring models, with special attention paid to machine learning approaches outperforming traditional models. We explore a recently proposed method called the PLTR model, which is a combination of machine learning and traditional logistic regression. In addition, we examine the models' performance and analyze the economic impact for different class weights. The main purpose of this paper was to identify the most effective and practical approach for credit scoring in the Swedish retail banking context. The findings suggest that the model that most accurately predicts defaults is the random forest, but at a high cost of interpretability due to the models' complexity. According to our findings, the optimal substitute for the random forest is a penalized logistic regression, as it compensates with interpretability, for slightly less accurate predictions.

Acknowledgements

We would like to express sincere gratitude towards our supervisor, Marcin Zamojski, for the valuable guidance and support throughout this project. This thesis would not have been the same without your feedback and honest opinions. We also want to thank the teachers at the University of Gothenburg for bringing us knowledge and inspiration throughout our studies at this school. Lastly, special thanks to Collector Bank for providing us with the data needed for the analysis in this thesis.

Contents

List of Tables

List of Figures

1 Introduction

The indebtedness in Swedish households has been rising sharply in recent years, driven partly by the increasing prevalence of consumption loans (Finansinspektionen 2022). In 2021, the total volume of consumer credit amounted to approximately four billion SEK, where unsecured loans accounted for 52 percent. These loans provide households with greater flexibility in spending, but they also carry significant future risks in interest payments and amortization. Especially during periods of escalating inflation, where households with significant debt face even greater vulnerability.

As demand for consumption loans increases, the market for loan brokers in Sweden has grown rapidly in the last few years (Finansinspektionen 2022). Consumers can apply for loans through intermediaries who connect them with various lenders. These brokers receive commissions from lenders for each approved loan. However, the Swedish Financial Supervisory Authority (FI) has analyzed this market and identified potential risks that could undermine consumer protection. Agreements between brokers and lenders sometimes include provisions that prevent lenders from directly obtaining necessary information from consumers for proper credit assessments. Additionally, there are instances where brokers impose requirements on lenders to approve a certain number of loans in order to receive more loan referrals. This creates incentives for banks and other lenders to grant loans to individuals who may not be able to repay them. As a result, it is essential for lenders to implement a credit scoring system that can accurately determine borrowers' ability to manage their debt and make timely payments (Finansinspektionen 2022). Such a system must be reliable, sustainable, and able to assess risks associated with different types of loans. By doing so, lenders can make informed decisions and promote financial stability for themselves and their clients.

Traditionally, credit scoring models in retail banking have relied on logistic regression, but recent developments in machine learning and AI have shown that ensemble methods such as random forest can outperform older methods in predicting default probability. However, such methods are often criticized for their lack of interpretability, which can create difficulties for financial regulators seeking to ensure that lenders are making sound and responsible decisions without discrimination (Dastile et al. 2020).

This thesis aims to investigate and compare different credit scoring models, including both traditional logistic regression and more advanced methods such as random forest and classification trees. In addition, we also use a recent method called Penalized Logistic Tree Regression (PLTR) proposed by Dumitrescu et al. (2022). We use a large dataset that contains extensive financial and personal information on approved loan applications at Collector Bank. The dataset is anonymized by excluding personally identifiable information in order to follow the credit scoring regulations in Sweden. The probability of default is estimated for three different dependent variables: total defaults within the whole dataset, defaults within 12 months, and defaults within 5 months. This is done in order to identify if the effect of the variables differs depending on the time before the loan defaults. Additionally, the models' performances are analyzed for various class weights. Examining the model's performance with different class weights allows us to evaluate its effectiveness in addressing the challenges posed by class imbalance. Furthermore, we evaluate the economic impact of each model by considering the effect of different class weights. By evaluating all the models' performance in terms of accuracy, interpretability, and other relevant factors, this study seeks to identify the most effective and practical approach for credit scoring in the Swedish retail banking context. Ultimately, the goal is to provide valuable insights and guidance for both lenders and regulators, helping to balance the needs for profitability and risk management.

Our findings are in line with previous literature provided by Brown & Mues (2012), Moscato et al. (2021), and Trivedi (2020). Without any class weight, the more advanced methods, such as the random forest, clearly outperform the more simple methods based on the logistic regression. In comparison to Dumitrescu et al. (2022), it can be inferred that the PLTR has comparable performance to penalized logistic regression, inferior performance compared to decision tree-based methods while outperforming the regular logistic regression. When adding different class weights for the models, the PLTR and the penalized logistic regression have more similar performance. The random forest still has the best performance, but not far away from PLTR and the penalized logistic regression. The suggested results from using class weights imply that lower class imbalance would result in predicting defaults more accurately, but at the same time denying more applicants that do not actually default (assuming the same cut-off strategy between different class imbalances). To summarize our findings, using penalized logistic regression with a class weight is an alternative to more advanced methods. The model shows a similar performance and it is easier to interpret, making it less difficult to follow the Consumer Agency Guidelines for credit scoring.

The thesis is organized as follows: Section 2 contains the literature review, which discusses previous studies on the subject of credit scoring. Section 3 explains and summarizes the data from Collector Bank that is used in the thesis. Section 4 contains the theory behind the used methods. In section 5, we explain the empirical strategy that we use, and section 6 shows the results and the discussion. Lastly, section 7 includes the conclusion and further research.

2 Literature review

In this section, we discuss previous literature on credit scoring, where we focus on six main fields; classification models and their performance, data processing and variable selection, data splitting and resampling, hyper-parameter optimization, ethical discussion and regulation of credit scoring in Sweden. The first four subsections compare different approaches in literature and their findings, whereas the last subsections discuss ethical issues and regulations in the Swedish credit market.

2.1 Introduction of models

Classification models predict the category or class to which a certain observation belongs, based on given variables. Among the models, one can find sophisticated machine learning approaches and simpler approaches such as a logistic regression.

The logistic regression is the model used to predict the probability that an observation belongs to a particular category. The model assumes that there is a linear relationship between the dependent and independent variables, requiring less computational power and allowing it to be trained on small datasets. This makes it a useful tool for situations where interpretability and transparency are important, such as in the case of financial risk analysis (Gareth et al. 2013).[1]

On the other hand of the complexity spectrum, classification methods such as tree-based models, are conventionally referred to as machine learning. These models are useful as they are flexible tools that can handle complex, non-linear relationships and efficiently handle high-dimensional datasets. Tree based models can be divided into two main categories: decision trees and ensemble methods such as random forests. A decision tree divides sample data into subsets based on features of observations, which creates a

[1] interpretability in this thesis is defined as how inputs relate to output. In more advanced approaches it is difficult to know the exact impact of a change in a variable on the outcome.

tree. Thus, it recursively splits the data based on the values of given features, where each node is a decision based on a specific feature and the leaf node represents the class of the dependent variable. Random forests have the same logic as decision trees, but are obtained by averaging multiple decision trees based on different subsets of data in order to reduce overfitting and variance (Gareth et al. 2013).

However, random forests are considered to be a "black box". The approach is complex and the results are hard to interpret which hinders the decision-making process. In order to address the disadvantages of random forests, Dumitrescu et al. (2022) propose a new method called penalized logistic tree regression (PLTR). This model uses information from multiple decision trees to improve the performance of a logistic regression. In that way, the PLTR model allows for non-linear effects that can arise in credit-scoring data while preserving the interpretability of logistic regression and obtaining similar performance to random forests.

Other machine learning approach include neural networks (NN), where the artificial neural network (ANN) is the most simple NN. This machine-learning method mimics the structure and function of the human brain. The networks can be thought of as a sequence of non-linear models generating regressors for the model further down the pipeline. Random forests are suited for structured datasets with many features, while ANNs are better for complex problems where traditional ML methods may not be effective (Gareth et al. 2013).[2]

2.2 Classification models and their performance

In practice, logistic regression is the most commonly used method for evaluating the credit-worthiness of borrowers. This is due to the simplicity and transparency in its predictions as well as a long history of use. Nonetheless, the more sophisticated machine learning models can be found to outperform the logistic regression (Dastile et al. 2020). In fact, empirical evidence has shown that machine learning models have superior performance compared to logistic regression when applied to credit scoring (Brown & Mues 2012, Trivedi 2020, Moscato et al. 2021, Kruppa et al. 2013, Dumitrescu et al. 2022).

[2] The most commonly used methods, NN included, are mentioned due to completeness. Although, we do not use NN as one of the models in our thesis.

Brown & Mues (2012) use Friedman's test in order to compare the so-called area under the curve score (AUC-score)[3] with the findings that random forests outperform other methods,[4] especially when faced with a large class imbalance in the dataset.[5] Since a single decision tree relies on a single split criterion to partition the data, it may be prone to overfitting and might not capture local characteristics that are not evident in the dataset as a whole. By combining multiple trees and looking at different subsets of data, random forest can capture the patterns and relationships that exist within subsets of data (Gareth et al. 2013). Random forests are therefore able to capture more complex and subtle relationships between the explanatory variables and the output variable. When comparing models and their AUC score, Gini index, false-positive rates and false-negative rates, the typical finding is that random forest have best the performance overall (Trivedi 2020, Moscato et al. 2021, Kruppa et al. 2013, Dumitrescu et al. 2022).

However, the supremacy of random forest is questioned by Baesens et al. (2003), Wang et al. (2011), and Lessmann et al. (2015). Baesens et al. (2003) find that several classification techniques that use machine learning algorithms have similar level of accuracy, effectiveness and overall performance. Specifically, support vector machines and neural networks yield the best performances for their datasets. Lessmann et al. (2015) update the study of Baesens et al. (2003) with more recently developed methods. Their overall result is that support vector machines and neural networks yield the best performance in terms of Pearson's correlation coefficient and the area under the ROC curve. However, other classification techniques provide performances that are quite competitive as well. In contrast, Wang et al. (2011) find that bagging together with decision trees is the most effective method with regards to average accuracy, false-positive rate, and false-negative rate.

[3] The Reciever Operating Characteristic curve (ROC curve) displays the true positive rate against the false positive rate for a binary classification model. The AUC-score assesses the discriminatory ability of the predictions, by representing the area under the ROC curve. (Gareth et al. 2013).

[4] Brown & Mues (2012) test the following methods; logistic regression, decision trees, neural networks, linear discriminant analysis, quadratic discriminant analysis, k-nearest neighbours, support vector machines.

[5] A large class imbalance occurs when the difference between the number of observations belonging to different classes in a dataset is substantially large.

2.3 Data processing and variable selection

One critical aspect of credit scoring is the inclusion of relevant variables that provide insight into a customer's creditworthiness. The data used for evaluation of credit scoring models can either be privately or publicly available. When evaluating the performance of credit scoring models, most studies use datasets that are publicly available (Trivedi 2020, Dumitrescu et al. 2022, Dastile et al. 2020, Lessmann et al. 2015, Wang et al. 2011). The publicly available datasets are adapted specifically for creating credit scoring models. As mentioned by Dastile et al. (2020), the most frequently used real-life publicly available datasets are the 'German' and 'Australian' credit datasets. Both of these datasets are publicly available at the University of California, Irvine Machine Learning repository (UCI). The 'German' Credit Risk dataset contains 1,000 entries with a total of nine categorical or symbolic variables. Each data point represents a person who took a loan from a bank. The 'Australian' dataset contains credit card applications. Names and personal information have been removed from both the 'German' and the 'Australian' datasets. The Australian dataset differs from the German dataset in that it originally contained missing values (5%). To handle this, the UCI preprocessed the data by replacing the missing values with the mode (for categorical variables) or mean (for continuous variables). In contrast, the German dataset did not have any missing values to begin with. Lessmann et al. (2015), Brown & Mues (2012) and Wang et al. (2011) use both datasets, while Trivedi (2020) only use the 'German' dataset.

There are also studies that use confidential data sources. Moscato et al. (2021) uses a dataset from a real lending institution, which includes approximately 877,000 observations and 151 features. In contrast to the 'German' and 'Australian' datasets, the dataset used by Moscato et al. (2021) is considerably larger, both in terms of observations and number of features.

The variables included in the most commonly used 'German' dataset are: age, gender, occupation, housing, saving accounts, checking amount, credit amount, duration, and purpose for applying for the loan.[6] Studies have shown that men have a higher default risk than women in various financial contexts, such as lending and credit. One reason for this is that men tend to exhibit riskier financial behaviors (Cigsar & Deniz 2018).

[6] Saving accounts are bank accounts with interest, where the goal is to only store money without transactions. Checking accounts are bank accounts for regular transactions such as purchases and bill payments.

If the applicant is married, a higher financial stability is expected. This is due to the fact that spouses often share financial responsibilities, which reduces the probability of default (Moscato et al. 2021). Homeownership status can also impact the probability of default, as homeowners generally have more financial stability due to the opportunity to build equity over time. Although, the financial stability is also dependent on if mortgages have fixed interest rates. According to Greenberg et al. (2019), comparing credit profiles of American renters and owners, individuals who have been approved for a mortgage are generally more creditworthy.

Trivedi (2020) argues that preprocessing of data is important for the performance of the models, this includes methods to handle missing values in the dataset. One solution, used by Moscato et al. (2021), is to remove variables that have a missing value percentage greater than 55%. The missing values that remain are then replaced with the variable medians. In line with Moscato et al. (2021), Trivedi (2020) uses a similar method for replacing missing values. But they replace missing values with the mean if the variable is numeric. If the variable is non-numeric, the missing values are replaced with the mode.

Some sources opt to include all available data. For instance Dumitrescu et al. (2022), Dastile et al. (2020) and Lessmann et al. (2015) do not use any specific feature selection method. Since they use datasets that are adapted for credit scoring, they choose to include all variables in the datasets. Moscato et al. (2021) investigate the correlation between variables in order to decide which variables to include. They do this based on a correlation matrix on the full dataset: the explanatory variables that have a too high correlation with each other are dropped. Trivedi (2020) uses different feature selection techniques to examine which variables have the largest effect on the probability of default. The feature selection techniques used by Trivedi (2020) are information-gain, gain-ratio, and Chi-square. Trivedi (2020) uses the top 15 variables that have the highest significance of the association between the dependent and independent variables in the 'German' dataset. The variables with the highest significance are the type of apartment, concurrent credits, gender and marital status, foreign worker, account balance, and payment status of the previous credits.

2.4 Data splitting and resampling

Typically, data is split into a training and test subsamples to enable machine learning approaches to learn patterns within the dataset and assess their ability to generalize to unseen data. Hence, the statistical learning method is trained to more accurately predict the outcome of new observations based on patterns observed in the training sample. This technique, known as data splitting, helps to prevent unnecessary biases and overfitting. Resampling methods such as cross-validation play a crucial role in this process by repeatedly drawing samples from the training data and refitting the model (Gareth et al. 2013).

Previous literature use k-fold cross-validation (explained in section 4.2), which is used to estimate the test error associated with the statistical learning method in order to review its performance (Trivedi 2020, Brown & Mues 2012, Lessmann et al. 2015, Moscato et al. 2021). The most commonly used number of folds are 5 or 10, since it leads to an intermediate level of bias (Gareth et al. 2013).[7]

2.5 Model choice and hyperparameter tuning

Hyperparameters are parameters of a model that are not estimated, for most methods they need to be set before training the model. The performance of a method is typically sensitive to the choice of hyperparameters. By finding the best set of hyperparameters the performance of a machine learning model can be drastically improved, both in terms of effectiveness and accuracy (Yang & Shami 2020).

The most widely used methods for tuning hyperparameters are manual search, grid search, and random search (Bergstra & Bengio 2012, Yang & Shami 2020). Of these, manual testing is a rudimentary way to tune hyperparameters. It is also inefficient and it requires a deep understanding of the algorithms. Manual search requires a lot of human effort in order to tune the parameters for larger datasets, where there is a greater number of hyperparameters to tune (Yang & Shami 2020, Probst et al. 2019). In contrast, grid search is a more computationally intensive method in comparison to manual search. It involves defining a grid of hyperparameters and evaluating the model performance for all possible combinations of hyperparameters that are specified. A limitation of the method is that it is time consuming due to it going through all possible combinations, and not always able to detect a global optimum of continuous parameters. Grid search is therefore

[7] Trivedi (2020) and Moscato et al. (2021) choose to use 10 fold cross-validation. Moscato et al. (2021) use 4-fold validation, whereas Trivedi (2020) use 3-fold validation and Brown & Mues (2012) use a 70:30 between the training and testing data. In contrast to earlier mentioned papers, Lessmann et al. (2015) use two folds instead of 10.

more useful for smaller numbers of categorical or discrete hyperparamaters(Yang & Shami 2020, Bergstra & Bengio 2012).

Finally, to optimize both continuous and discrete hyperparameters, random search can be used. Random search is more efficient than grid search because it does not need to examine every possible combination of hyperparameters. Instead, it samples random combinations of hyperparameters from the search space. The hyperparameters are drawn randomly, for example from a uniform distribution given pre-specified bounds. One drawback of the random search is that it is based on a random sampling, so there is no guarantee that it can find the optimal set of hyperparameters. Therefore, it is possible that random search may overlook some areas of the hyperparameter space that are important for achieving the best performance of the model. Yang & Shami (2020) find that the most important parameters to tune for random forests are: the number of estimators, maximum depth, minimum sample split, minimum sample leaf, and maximum features. For the decision trees: criterion, maximum depth, and minimum sample split are the most important parameters to tune.[8]

2.6 Ethics of credit scoring

Use of credit scoring models leads to a number of ethical issues, including concerns about fairness, transparency, privacy and potential discrimination. One the one hand, new data sources and advanced analysis techniques are being used to improve consumers' access to financial services, while making the process more efficient and less expensive. A more accurate and fair credit scoring method could benefit both the customers and the banks, with customers potentially receiving fairer interest rates and banks potentially increasing their lending and profits (Gutiérrez-Nieto et al. 2016). However, like any innovation, these developments can have unintended negative effects and raise ethical concerns about introducing a potential bias in lending decisions (Purda & Ying 2022).

Allowing borrowers to provide additional demographic details lending platforms may lead to biased lending decisions based on gender, ethnicity, appearance, or physical char-

[8] The number of estimators is the number of trees in the forest. Maximum depth refers to the maximum number of splits allowed for in individual decision trees within the forest. The minimum sample split is the minimum number of samples required to split an internal node. Minimum sample leaves are the minimum number of samples required to be at a leaf node and maximum features are the number of features to consider when looking for the best split.

acteristics. Studies indicate that such bias can be both unintentional and intentional, resulting in inappropriate and systematically biased lending decisions (Chen et al. 2017, Duarte et al. 2012). Moreover, there is a risk that credit scoring may perpetuate existing social and economic inequalities. For example, credit scoring algorithms may be biased against certain groups, such as low-income or minority populations, who may have less access to credit and other financial services. This bias can perpetuate a cycle of poverty and exclusion from the financial system (Verma 2019, Barocas & Selbst 2016). Another ethical concern is the potential for credit scoring to infringe on consumer privacy. Credit scoring companies may collect and use large amounts of personal data, such as financial and demographic information, to calculate credit scores. Consumers may not be aware of how their personal data is being used and may not have control over how it is shared or sold to other companies (Commission et al. 2019, Solove 2006).

2.7 Regulation for credit scoring in Sweden

Credit scoring has become an increasingly important tool for lenders in Sweden as they seek to assess the creditworthiness of potential borrowers. While credit scoring can make the lending process more efficient and less expensive, it also raises concerns about fairness, transparency, and privacy. In response, regulators in Sweden have taken steps to promote greater transparency and accountability in the credit scoring process. One important development in credit scoring regulation in Sweden is the introduction of the General Data Protection Regulation (GDPR) in 2018. The GDPR provides consumers with greater control over their personal data and requires companies to obtain explicit consent before collecting and using the data (Regulation 2018, Goddard 2017). This has important implications for credit scoring, as lenders must now ensure that they have obtained consent from consumers to collect and use their personal data in their models.

In addition to the GDPR, Sweden has implemented a number of other measures to regulate credit scoring. For example, Swedish Financial Supervisory Authority (Finansinspektionen) has published guidelines on credit scoring that provide recommendations for lenders on how to ensure that their credit scoring models are fair, transparent, and accurate (Konsumenternas 2022).[9] The guidelines recommend that lenders should use data

[9] The guidelines include that the models should take into account personal circumstances such as family composition, housing sitation and debts. In addition, the lender should also take age into account. It is important that young people do not risk taking on large debts early in life. In conclusion, the individual should not risk suffering payment problems or financial vulnerability during the term of the

that is relevant and up-to-date. In addition, lenders should be aware of any systemic biases that may exist in the data, such as discrimination against certain ethnic or socioeconomic groups, and take steps to eliminate or minimize those biases

Another important development in credit scoring regulation in Sweden is the introduction of the Credit Information Act (Kreditupplysningslagen) in 2018. This law regulates the use of credit information and credit scoring in Sweden and provides consumers with the right to access their credit reports and correct any errors or inaccuracies in the reports. The law also requires credit scoring companies to be transparent about how they collect and use credit information, and to provide consumers with information on how to dispute errors in their credit reports (*Kreditupplysningslag (1973:1173)* 1973)

3 Data

3.1 Introduction of the sample

The underlying data is provided by Collector Bank. Collector Bank is a digital bank specializing in financial services for enterprises and private customers and complements traditional banks. Their private segment amounts to 31% of their total loan portfolio where they offer both uncollateralized customer loans and credit cards. During our sample period, Collector Bank lowered its cut-off strategy.[10] This means that the current lending policy makes them grant fewer loans, compared to their prior policy. Thus, this is a factor that may affect our results.

In general, Collector Bank's loan portfolio shows a higher proportion of female customers and the average customer is of middle age. The income distribution among Collector Bank customers mirrors the general Swedish population. However, a larger population of low-middle-income households is apparent among the bank's customers than in the overall Swedish population. The underlying cause is that their portfolio has a smaller proportion of customers with lower or non-existing income, which arises from Collector Bank having minimum income requirements. Overall, the debt ratios of Collector Bank customers tend to be high, where their debts typically consist of various types of commer-

credit (Konsumenternas 2022)

[10] A cut-off strategy refers to a threshold based on a score used to determine whether an applicant is qualified to be granted a loan or not. The applicant will be denied if the applicant has a higher score than a certain threshold. The score, in this case, is the probability of default.

cial loans rather than mortgages. The explanation for this is that a significant proportion of customers in the Collector Bank's lending portfolio live in rental properties, which is greater than the proportion of renters in the general population of Sweden.

3.2 Ethical overview of the sample

We use a large dataset that contains extensive financial and personal information on approved loan applications at Collector Bank, where we carefully choose variables from Collector Bank's database based on previous literature. It is essential that we follow the credit scoring regulations in Sweden, called GDPR, so that ethical issues such as fairness, privacy, and potential discrimination are avoided. Consequently, the dataset is anonymized by excluding personally identifiable information such as the Swedish personal number and names of the applicants. This is to prevent the possibility of linking specific individuals to their personal information given at the time of the application. In addition, the dataset includes postal codes in order to be able to extend the dataset by matching observation with the Swedish Police's database on 'exposed areas'. In a 2021 report, the Swedish Police assigns each postal code a risk category; risk areas, especially exposed risk areas, and areas in danger of becoming a risk area.

The resulting dataset consists of loans that were paid out at the end of 2018 until the beginning of 2023, and it includes only Swedish applicants.[11] The dataset has a total of 168,113 rows, but some of the rows include both the applicant and the co-applicant.

3.3 Data processing

Initially, the resulting dataset is subjected to a filtering procedure. As previously mentioned we exclusively include Swedish applicants, and the reason behind this is that the bank stopped its lending operations outside of Sweden in 2020. Thereafter, denied applications are excluded resulting in a dataset that only consists of loans that have been granted and paid out. To avoid duplicates among the loans with a co-applicant, we exclude the applicant with a higher default probability. This is achieved through sorting the data based on an external risk forecast measure used by the bank.[12] Subsequently, the duplicate with the highest risk forecast is eliminated from the dataset.

[11] A Swedish applicant is a person with a Swedish personal number.

[12] The risk forecast measure is provided by UC, and it measures the risk involving the applicant receiving a payment report request within 12 months.

The overall dataset consists of loans that are active, have defaulted, or have been fully re-paid. We focus on three cases when evaluating the probability of defaults; total defaults, defaults within 12 months, and defaults within 5 months, which results in 3 different datasets. The "total default" dataset consists of almost 42,000 observations. This dataset only consists of applicants with an outcome; either they have defaulted or the loan is fully repaid, resulting in active loans being dropped. The second dataset is the "12-month defaults", which contains applicants that have an outcome at or after 12 months. This means that all applicants that are still active within the last 12 months are dropped, resulting in approximately 62,400 observations. Lastly, the third dataset is the "5-month defaults" with almost 67,600 observations.[13] This dataset contains the applicants that have an outcome at or after the last 5 months, resulting in loans that are active within the last 5 months being dropped. We are dropping the active loans within each dataset since the survival time is unknown, which are the observations that are beyond the time of the last observation. Removing loans with unknown survival time is referred to as "right-censoring" in literature. We acknowledge that right censoring can lead to biased results since the ability to estimate the precise outcome is limited.

Categorical variables in the dataset are converted to continuous variables or dummies depending on the categories (see Appendix 8). Classification models are able to handle both categorical and continuous variables, but logistic regression models can only handle continuous variables or dummies. Therefore, converting the categorical variables is necessary in order to evaluate all models.

Table 1 shows both the number of missing values and the percentage of missing values for each variable in the dataset. The table presents either none or a minimum amount of missing values for the majority of the variables, as we removed the variables that exhibited a proportion of missing values exceeding 30% from the dataset. Moreover, the missing values shown in Table 1 are filled with values using the k-th nearest neighbor algorithm (see Section 4.1 for details on the algorithm) with k=5.

[13] The defaults within 5 months include "straight rollers", which are the applicants that have pending payments on their loans until they default.

Table 1: Percentage of Missing Values for each variable

Number of missing values	5 month defualts	12 month defaults	Total defualts
Age	0 (0.0%)	0 (0.0%)	0 (0.0%)
No.Cars	0 (0.0%)	0 (0.0%)	0 (0.0%)
No.Children	0 (0.0%)	0 (0.0%)	0 (0.0%)
No.Adults_inhousehold	0 (0.0%)	0 (0.0%)	0 (0.0%)
CivilStatus	0 (0.0%)	0 (0.0%)	0 (0.0%)
Occupation	0 (0.0%)	0 (0.0%)	0 (0.0%)
UC_AnnualIncome	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_CreditusedApartment	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_CreditusedHouse	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_CreditusedBlanco	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_NumberOfCreditors	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_NumberOfReportRequests	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_RiskForecast	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_CreditUsedRevolving	29 (0.0%)	24 (0.0%)	17 (0.0%)
UC_CreditUsedPartPayment	29 (0.0%)	24 (0.0%)	17 (0.0%)
Role	0 (0.0%)	0 (0.0%)	0 (0.0%)
NumberOfApplicants	0 (0.0%)	0 (0.0%)	0 (0.0%)
ApplicationAmount	0 (0.0%)	0 (0.0%)	0 (0.0%)
RepaymentMonths	0 (0.0%)	0 (0.0%)	0 (0.0%)
ProbabilityOfDefault	0 (0.0%)	0 (0.0%)	0 (0.0%)
Municipal group 2023	0 (0.0%)	0 (0.0%)	0 (0.0%)
Exposed_areas	3 (0.0%)	2 (0.0%)	1 (0.0%)
granted_loweramount	0 (0.0%)	0 (0.0%)	0 (0.0%)
gender	0 (0.0%)	0 (0.0%)	0 (0.0%)
Debtratio_blanco	31 (0.0%)	26 (0.0%)	19 (0.0%)
Debtratio_morgage	31 (0.0%)	26 (0.0%)	19 (0.0%)
New_blancodebt	0 (0.0%)	0 (0.0%)	0 (0.0%)
stated_diff_income	0 (0.0%)	0 (0.0%)	0 (0.0%)
Default	0 (0.0%)	0 (0.0%)	0 (0.0%)

3.4 Descriptive statistics

3.4.1 Descriptive statistics of the independent variables

For each of the three datasets, specific descriptive statistics have been established. Table 2 contains the number of observations, the mean, the median, and lastly the standard deviation.

Table 2: Descriptive Statistics of independent variables

Table 2 consists of descriptive statistics for the total default dataset, 12-month default dataset, and 5-month default dataset. The description of variables is presented in Appendix 8. The financial variables; Annual income, Mortgages, blanco debt, credit card amounts, and application amount, are expressed in Million SEK.

	count	mean	std	Median
Age	41956	45.08	12.73	45.0
No.Cars	41956	0.13	0.33	0.0
No.Children	41956	0.54	0.88	0.0
No.Adults_inhousehold	41956	0.72	0.87	0.0
CivilStatus	41956	2.01	1.85	2.0
Occupation	41956	3.08	0.56	3.0
UC_AnnualIncome	41939	0.40	0.18	0.36
UC_CreditusedApartment	41939	0.15	0.54	0.0
UC_CreditusedHouse	41939	0.32	0.86	0.0
UC_CreditusedBlanco	41939	0.35	0.25	0.33
UC_NumberOfCreditors	41939	5.41	2.68	5.0
UC_NumberOfReportRequests	41939	7.7	6.5	6.0
UC_RiskForecast	41939	4.41	3.95	3.3
UC_CreditUsedRevolving	41939	0.03	0.05	0.02
UC_CreditUsedPartPayment	41939	0.02	0.06	0.0
Role	41956	0.96	0.19	1.0
NumberOfApplicants	41956	1.05	0.22	1.0
ApplicationAmount	41956	0.18	0.16	0.1
RepaymentMonths	41956	118.17	51.13	120.0
Municipal group 2023	41956	5.56	2.25	6.0
Exposed_areas	41955	0.42	0.87	0.0
granted_loweramount	41956	0.22	0.41	0.0
gender	41956	0.39	0.49	0.0
Debtratio_blanco	41937	0.87	0.54	0.89
Debtratio_morgage	41937	0.66	2.26	0.0
New_blancodebt	41956	0.07	0.25	0.0
stated_diff_income	41956	0.54	0.5	1.0

(a) Total defaults

	count	mean	std	Median
Age	62403	45.86	12.6	46.0
No.Cars	62403	0.12	0.32	0.0
No.Children	62403	0.52	0.88	0.0
No.Adults_inhousehold	62403	0.69	0.87	0.0
CivilStatus	62403	1.99	1.83	2.0
Occupation	62403	3.09	0.55	3.0
UC_AnnualIncome	62379	0.40	0.18	0.37
UC_CreditusedApartment	62379	0.15	0.54	0.0
UC_CreditusedHouse	62379	0.33	0.87	0.0
UC_CreditusedBlanco	62379	0.36	0.26	0.34
UC_NumberOfCreditors	62379	5.63	2.81	5.0
UC_NumberOfReportRequests	62379	7.31	6.33	6.0
UC_RiskForecast	62379	4.41	4.1	3.2
UC_CreditUsedRevolving	62379	0.04	0.06	0.02
UC_CreditUsedPartPayment	62379	0.02	0.06	0.0
Role	62403	0.94	0.24	1.0
NumberOfApplicants	62403	1.08	0.27	1.0
ApplicationAmount	62403	0.19	0.17	0.13
RepaymentMonths	62403	121.09	50.27	120.0
Municipal group 2023	62403	5.54	2.26	6.0
Exposed_areas	62401	0.41	0.86	0.0
granted_loweramount	62403	0.22	0.41	0.0
gender	62403	0.39	0.49	0.0
Debtratio_blanco	62377	0.88	0.55	0.9
Debtratio_morgage	62377	0.65	2.27	0.0
New_blancodebt	62403	0.07	0.26	0.0
stated_diff_income	62403	0.55	0.5	1.0

(b) 12 month defaults

	count	mean	std	Median
Age	67577	45.93	12.54	46.0
No.Cars	67577	0.13	0.34	0.0
No.Children	67577	0.53	0.88	0.0
No.Adults_inhousehold	67577	0.7	0.87	0.0
CivilStatus	67577	2.0	1.81	2.0
Occupation	67577	3.09	0.54	3.0
UC_AnnualIncome	67548	0.41	0.18	0.37
UC_CreditusedApartment	67548	0.15	0.56	0.0
UC_CreditusedHouse	67548	0.34	0.90	0.0
UC_CreditusedBlanco	67548	0.36	0.26	0.34
UC_NumberOfCreditors	67548	5.64	2.82	5.0
UC_NumberOfReportRequests	67548	7.33	6.39	6.0
UC_RiskForecast	67548	4.48	4.19	3.2
UC_CreditUsedRevolving	67548	0.04	0.06	0.02
UC_CreditUsedPartPayment	67548	0.02	0.06	0.0
Role	67577	0.93	0.25	1.0
NumberOfApplicants	67577	1.08	0.28	1.0
ApplicationAmount	67577	0.19	0.17	0.13
RepaymentMonths	67577	121.43	50.19	120.0
Municipal group 2023	67577	5.55	2.26	6.0
Exposed_areas	67574	0.41	0.86	0.0
granted_loweramount	67577	0.22	0.42	0.0
gender	67577	0.39	0.49	0.0
Debtratio_blanco	67546	0.89	0.55	0.91
Debtratio_morgage	67546	0.67	2.36	0.0
New_blancodebt	67577	0.07	0.26	0.0
stated_diff_income	67577	0.55	0.5	1.0

(c) 5 month defaults

Looking at the data, the average applicant is 45 years old and is married or cohabitating, with an annual income of approximately 400,000 SEK. In addition, the applicant has a permanent occupation of more than 6 months and they live in a commuter municipality near a big city. Furthermore, most of the applicants are women (60%). Only 4% of the applicants are co-applying to the loan, whereas the rest are main applicants.

3.4.2 Descriptive statistics dependent variables

As mentioned, we have three different datasets (see section 3.3) in which the dependent variable, default, is defined differently.

Table 3: Descriptive Statistics of Dependent variables

	count	mean	std
Total defaults	41956	0.182	0.39
12m Defaults	62403	0.059	0.24
5m Defaults	67577	0.014	0.12

Table 3 contains descriptive statistics of these dependent variables. "Total defaults" is represented by the defaulted applicants within the "total default" dataset. The average percentage that defaults within this dataset is 18%. Correspondingly, the "12m defaults" refers to the applicants who have defaulted within 12 months, which accounts for 6%. Lastly, the "5m defaults" is the applicants that have defaulted within the last 5 months, representing 1.4% of the "5m default" dataset. It is noticeable that the defaulted applicants represent a small fraction within each dataset. Thus, all datasets exhibit a significant class imbalance, especially the 12-month defaults and 5-month defaults.

3.5 Splitting the data into train and test samples

In order to control for overfitting, we split the data into two sub-samples; training and test. The training subsample is used to train the models, whereas the five-fold cross-validation is used in order to validate the results from training the models. Thereafter, the models are tested on the test subsample.

Initially, the dataset is sorted according to the disbursement date of the loan. Sorting the data based on time is important in order to avoid data leakage. It occurs when information from the test sample is inadvertently used to train the model. Given events and movement conditions that affect the whole population, the probability of default is inherently dependent on time, and hence, the issue of data leakage is a fact. Consequently, if the data is not sorted by time, there is a risk that the model recovers time patterns from future data that it should not have access to, leading to misleading results during evaluation. Sorting the data by time ensures that the model is only trained on past data which is available at that time, then evaluating on "future" data. This is also known as pseudo-out-of-sample analysis.

After sorting based on time, we do the data splitting. We set aside 20% of the most recent observations for the test set so that the remaining data is used to train the model. Since we have three different datasets, there are slightly different timespans for our training and test samples. These are shown in Table 4.

Table 4: Time periods for each sample

Datasets	Training set	Test set
Total defaults	15/5/2019 - 20/8/2021	20/8/2021 - 1/2/2023
12m defaults	15/5/2019 - 2/9/2021	2/9/2021 - 1/2/2023
5m defaults	15/5/2019 - 2/11/2021	2/11/2021 - 1/2/2023

4 Theory and method

4.1 Missing values and K-nearest neighbour

In order to handle the missing values without removing them from the data, there are two general approaches. Firstly, predictive models such as tree-based techniques can specifically account for missing data. Alternatively, the missing data can be imputed using the information within the data to estimate values based on other predictors (Kuhn et al. 2013).

One popular imputation technique is the k-nearest neighbour (kNN), which we use in this thesis. kNN finds observations within the dataset that are "closest" to the missing values and averages these nearby points to fill the missing values.

$$kNN = \frac{1}{N} \Sigma X_{i \neq j} \tag{1}$$

Given a positive integer k and a missing value for variable x within the dataset, the k-nearest neighbor classifier first identifies the k points in the sample data that are closest to this missing value based on variables without missing values, represented by N. It then averages the values of N, to fill in the missing value (Kuhn et al. 2013). The advantage of the kNN technique is that the imputed data is confined to be within the range of the dataset values. However, it can be time-consuming as the entire dataset is usually required every time a missing value needs to be imputed (Kuhn et al. 2013).

4.2 Cross-validation

One of the most commonly used re-sampling methods is cross-validation (refer to Section 3.5 for the data splitting process in this thesis). k-fold cross-validation involves dividing the data into k equally sized subsets (or folds). Then, $k-1$ folds are used for training the model and one fold is used for validation. The process is then repeated k times and the average performance across all k validation sets are then computed (Gareth et al. 2013). If five-fold cross-validation is used, one of the groups represents the validation group and the remaining four are set for training the model. Performance measures are computed on the observations in the held-out fold. This procedure is repeated k times and each time, a different group of observations is treated as a validation sample. The process results in k estimates of the MSE and the k-fold cross-validation estimate is computed by averaging these values:

$$CV_{(k)} = \frac{1}{k}\Sigma_{i=1}^{k} MSE_i. \tag{2}$$

However, there is a bias-variance trade-off associated with the choice of k. This means that decreasing k leads to overfitting and high biases, as the model is trained on a small set of data. Whereas, increasing k could lead to lower biases and reduce overfitting, but higher variance. This is because more folds may result in the models not capturing important patterns, leading to higher variance and making the performance of the models poorer. According to Gareth et al. (2013), the choice of $k = 5$ or $k = 10$ leads to an intermediate level of bias.

4.3 Classification Methods

Prediction of the probability of default can be accomplished using a number of techniques. Among these are classification models, for example, logistic regression, decision trees, random forests, linear discriminant analysis, and, or kNN (Gareth et al. 2013). The theory behind the methods that we use is carefully explained in this section. The methods used are logistic regression, penalized logistic regression, PLTR, decision trees, and random forests.

4.3.1 Logistic regression

Logistic regression is a linear model used to estimate the probability of a dependent variable Y belonging to a particular binary category. However, since a linear regression can potentially produce negative outcomes, it is necessary to transform the predictions of logistic regression into valid probabilities. To achieve this, logistic regression employs a specific function called the logistic function. This function maps the linear combination of the independent variables, denoted as X, into a range between 0 and 1. By applying the logistic function to the linear predictions, we obtain valid probabilities, $p(X)$, representing the likelihood of the dependent variable falling into a particular category.

To transform the predictions of logistic regression the following logistic function is used:

$$Pr(y_i = 1|x_i) = F(\eta(x_i; \beta)) = \frac{1}{1 + exp(-\eta(x_i; \beta))}, \tag{3}$$

where $F(.)$ is the logistic cumulative distribution functuon and $\eta(x_i; \beta)$ is the so-called index function defined as:

$$\eta(x_i; \beta) = \beta_0 + \Sigma_{j=1}^{p} \beta_j x_{i,j}, \tag{4}$$

where $\beta = (\beta_0, \beta_1, ..., \beta_p)$ is an unknown vector of parameters. The estimator $\hat{\beta}$ is obtained by maximizing the log-likelihood function. The intuition behind using maximum likelihood is as follows: we seek estimates for β_p such that the predicted probability of default, $Pr(y_i = 1|x_i')$, for each individual corresponds as closely as possible to the individual's observed default status. We try to find beta such that plugging these estimates into the model for $Pr(y_i = 1|x_i)$ yields a number close to one for the individuals who default, and zero for the individuals who do not default. This can be formalized using the likelihood function presented below:

$$\mathcal{L}(y_i; \beta) = \Sigma_{i=1}^{n} \Big\{ y_i \log\{F(\eta(x_i; \beta))\} + (1 - y_i) \log\{1 - F(\eta(x_i; \beta))\} \Big\}, \tag{5}$$

where x_i is the vector of predictor variables for the ith observation in the training dataset, $y_i = 0$ is when the customer defaults and $y_i = 1$ represent the non-defaults.

A penalized logistic regression is when a penalty term is added to the criterion function. The aim of penalization is to improve prediction performance by balancing the fit of the data and the stability of the estimates. The two best-known penalty terms are the ridge and lasso, however, this thesis only focuses on the lasso penalty (Gareth et al. 2013). The Lasso penalty uses the sum of the absolute values of the parameters and shrinks the coefficients toward zero. The Lasso penalty term is given by:

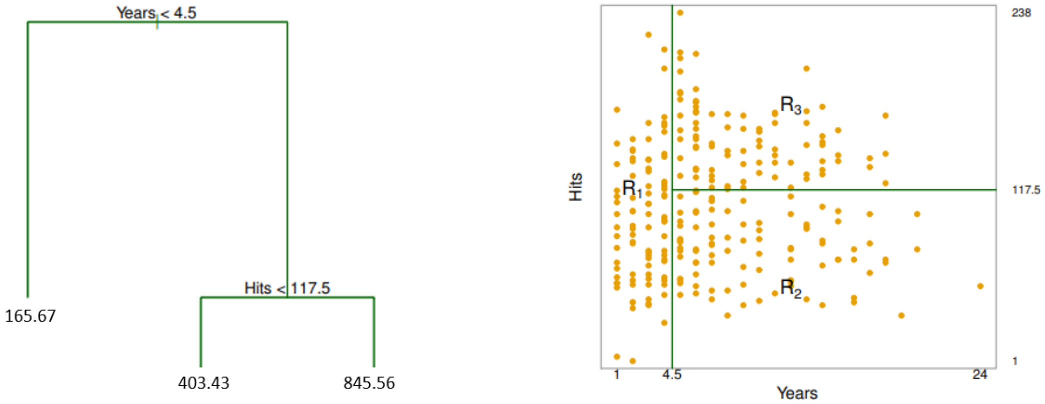
$$Lasso = \lambda P(\Theta) = \lambda \Sigma_{j=1}^{p} |\beta_j|, \tag{6}$$

where $P(\Theta)$ is the penalty term and λ is a tuning parameter that controls the intensity of the penalty.

4.3.2 Decision trees

The second method we consider are decision trees. Tree-based methods are a type of classifier that can be used for both regression (called regression trees) and classification (called classification trees). Tree-based methods involve segmenting the predictor space into simple regions. Since the set of splitting rules used to segment the predictor space can be graphically represented by a tree, these types of approaches are known as decision tree methods (Gareth et al. 2013).

Figure 1: Illustration of a decision tree



(a) Decision tree

(b) Regions of predictor spaces

Figure 1a illustrates a decision tree for predicting the salary (expressed in thousands of dollars) of a baseball player. The tree has two internal nodes and three terminal nodes, or leaves. Figure 1b illustrates the three-region partition for the observations in the dataset from the decision tree.

To give an example, consider the case of a baseball player's salary in Gareth et al. (2013). The salary is based on the number of years that the baseball player has played in the major leagues and the number of hits that the player has made in previous years. Figure 1a shows the decision tree which consists of a series of splitting rules. The top first split assigns observations having less than 4.5 years played in the major leagues to the left branch. The model predicts that these players have an average salary of approximately 165,000 dollars. Looking at Figure 1b, these players are represented by the first region R_1. Players that have played equally or at least 4.5 years in the major league are assigned to the right branch. These players are further grouped by the number of hits made in previous years. The players with less than 117 hits are assigned to the left side represented by the second region (R_2) and those above (or equal to) 118 are assigned to the right side representing the third region (R_3). Thus, the tree stratifies or segments the players into three regions of predictor space. In tree analogy, these regions are known as terminal nodes or leaves of the tree. The two points within the tree where the predictor space is split are referred to as internal nodes, which in this case are represented by the cut-off rules. The segments of the tree that connect the nodes are referred to as branches.

Individual decision trees allow for a lot of interpretability. In the example above, the number of years playing in the major league is the most important variable when it comes to predicting the salary as this is the first split. The explanation for this is that those who are less experienced earn lower salaries. The number of hits does not affect the salaries for the less experienced players, which is the case for the players that have more experience. Here, the players that have more hits in previous years tend to have higher salaries (Gareth et al. 2013).

The main difference between regression trees and classification trees is that regression trees are used to predict a quantitative response variable, whereas classification trees predict a qualitative response variable. For a regression tree, the predicted response for an observation is given by the mean response of the training observations that belong to the same terminal nodes. In contrast, the classification tree uses the majority rule for predicting. Thus, both the class prediction corresponding to a particular terminal node region and the class proportions among the training observations that fall into that region is of interest for a classification tree. However, the task regarding building a classification tree is similar to building a regression tree. The so-called recursive binary splitting is typically used to grow both trees. This is a top-down, greedy approach. It is ‘top-down’ because it begins at the top of the tree and then successively splits the predictor space and each split is indicated via two new branches further down in the tree. The decision tree algorithm is considered ”greedy” because it selects the best split at each step based only on the available predictors, without considering future steps. This approach may not always produce the optimal tree, but it is computationally efficient and tends to work well in practice for most datasets (Gareth et al. 2013).

The predictor X_j and threshold t for recursive binary splitting in a classification tree are selected to maximize the reduction in classification error rate. This is done by considering all possible combination values of X_j and t and choosing the split that yields the greatest reduction, as measured by the criterion. The classification error rate is the fraction of the training observations in that region that do not belong to the most common class. This can be given by the following equation.

$$E = 1 - \max_{k}(\hat{p}_{mk}), \tag{7}$$

where $\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from kth class. However, according to Gareth et al. (2013), classification error has some drawbacks as it is not sufficiently sensitive for tree-growing. Therefore, other measures such as the Gini index and entropy are preferable in practice. Firstly, the Gini index is defined by:

$$G = \Sigma_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{8}$$

which is a measure of total variance across the K classes. The Gini index takes on a small value if all of the $\hat{p}_{mk}$ are close to zero or one. For this reason, the Gini index is referred to as a measure of node purity—a small value indicates that a node contains predominantly observations from a single class. Alternatively, one can use entropy to grow a tree, which is defined as:

$$D = -\Sigma_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk}), \tag{9}$$

where $\hat{p}_{mk}$ lies between 0 and 1, which means that $0 \leq -\hat{p}_{mk} log(\hat{p}_{mk})$. Thus, entropy takes on a value near zero if $\hat{p}_{mk}$ are near zero or near one. By minimizing impurity, both algorithms aim to create subsets that are as homogeneous as possible with respect to their class labels.

If allowed to grow a lot, a tree can fully explain the training set, i.e., assign each observation to its own terminal node. Then, the tree might be too complex and likely to overfit the data, leading to poor test set performance. Hence, a smaller tree with fewer splits might lead to higher variance, but better interpretability and smaller bias. A way of achieving this is to prune the tree. This strategy involves growing a large tree and then pruning it back in order to obtain a smaller subtree. The goal is to obtain a subtree that leads to the lowest test error rate and limits overfitting (Gareth et al. 2013). Additionally, the problem of overfitting can be reduced by carefully choosing hyperparameters (see Section 2.5). Such hyperparameters are the criterion, maximum debt of the tree, the minimum number of samples required to split an internal node, the minimum number of samples required to be at a leaf node, the number of features to consider when looking for the best split, the maximum number of leaves and the quality of the split.

Decision trees for both regression and classification have numerous advantages over more classical approaches such as linear regressions or other classifiers such as logistic regressions. The advantage is that decision trees allow for capturing non-linear relationships. It includes simplicity and interpretability, which make them useful for tasks where it is important to understand the reasoning behind a decision. They can also handle qualitative predictors without the need for dummy variables, and they can be easily visualized. However, decision trees do not always have the highest prediction accuracy compared to other classifiers, which may limit their usefulness. Additionally, the trees can be non-robust, which means that a small change in the data can cause a large change in the final estimated tree. However, by aggregating many decision trees using methods such as bagging, random forest, and boosting, the predictive performance of the trees can be substantially improved (Gareth et al. 2013).

4.3.3 Random Forest

In order to reduce the variance of decision trees, one can use Bagged (Bootstrap Aggregation) trees. It involves creating multiple decision trees on random subsets of the training data. Each tree is grown independently, with no pruning, and all the trees are then combined to produce a final prediction. The combination is achieved by averaging or taking the majority vote of the predictions of the individual trees, resulting in reduced sensitivity to outliers and noise in the data.

While bagged trees help reduce variance, they still exhibit some correlation among the trees. Random forests, a variant of bagged trees, address this issue by introducing randomization during the tree-building process. Rather than considering all predictors at each split, only a random subset of predictors is utilized. This results in each tree being built with a different subset of predictors, effectively reducing the correlation between the trees (Gareth et al. 2013).

Random forests build multiple decision trees on subsets of the training data that are created by random sampling with replacement. As the trees are built and each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors. A new sample of m predictors is taken at each split, where the most commonly used number of predictors to include is $m \approx \sqrt{p}$ (Gareth et al. 2013).

The motivation for using only a subset of predictors is to avoid having all trees rely heavily on a single strong predictor, which would result in highly correlated predictions. Averaging many highly correlated predictions would not substantially reduce variance compared to a single tree. By considering only a subset of predictors at each split, random forest forces the trees to use different predictors, "decorrelating" them and resulting in a less variable and typically more reliable average of predictions. On average, the subset of predictors considered at each split excludes the strong predictor, which increases the chances of the other predictors being used (Gareth et al. 2013).

Random forests introduce additional hyperparameters compared to decision trees. These include the number of trees in the forest, the use of bootstrapping, and accuracy estimation using out-of-bag samples. These hyperparameters provide flexibility in controlling the ensemble's behavior and improving generalization (Gareth et al. 2013).

4.4 Penalized Logistic Tree Regression

The Penalized Logistic Tree Regression model (PLTR) is an extension of logistic regression that incorporates non-linear effects of predictor variables through decision-tree structures. The goal of the model is to improve credit scoring by better predicting the probability of default, while still being easy to interpret.

PLTR is a logistic regression model that includes explanatory variables based on single and two-step trees. The first step in constructing a PLTR model is to capture the univariate threshold effects by creating p decision trees (one for each prediction) with only one split to obtain these threshold effects. New columns of dummy variables are then created for every predictive variable p, where each observation gets a value of 0 or 1 depending on which node they belong to. In the second step, we use decision trees with two splits to capture bivariate threshold effects. Each bivariate decision tree generates at least three binary variables, each associated with a terminal node. This step ensures that all variables are combined at least once. We save these dummy variables representing the combinations as new columns in our dataset. Finally, the logistic regression with univariate and bivariate threshold effects has the following form:

$$P(y_i = 1|V_{i,1}^{(j)}, V_{i,2}^{(j,k)}; \theta) = \frac{1}{1 + exp[-\eta(V_{i,1}^{(j)}, V_{i,2}^{(j,k)}; \theta)]}, \tag{10}$$

where

$$\eta(V_{i,1}^{(j)}, V_{i,2}^{(j,k)}; \theta) = \beta_0 + \sum_{j=1}^{p} \alpha_j x_i + \sum_{j=1}^{p} \beta_j V_{i,1}^{(j)} + \sum_{j=1}^{p-1} \sum_{k=j+1}^{p} \gamma_{j,k} V_{i,2}^{(j,k)}. \tag{11}$$

$\theta = (\beta_0, \alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_p, \gamma_{1,2}, \ldots, \gamma_{p-1,p})$ are the set of parameters to be estimated, where β_j are the coefficients for the one-step dummies, and γ_j are the coefficients for the two-step dummies. The length of θ depends on the number of predictive variables, p. For example, when p=10, this leads to a total number of m=100 univariate and bivariate threshold effects in the regression.

To prevent overfitting issues with a large number of predictors, one can rely on penalization for both estimation and variable selection. In the PLTR method, a penalty term is added to the log-likelihood which results in the following expression:

$$L_p(R_{i,1}^{(j)}, V_{i,1}^{(j,k)}, V_{i,2}^{(j,k)}; \Theta) = -L(R_{i,1}^{(j)}, V_{i,1}^{(j,k)}, V_{i,2}^{(j,k)}; \Theta) + \lambda P(\Theta). \tag{12}$$

The adaptive Lasso estimator, proposed by Zou in 2006, is used by Dumitrescu et al. (2022) instead of the regular Lasso estimator. They use this penalty term due to its oracle property. This means that the adaptive Lasso applies a stronger penalty to coefficients that have large magnitudes while reducing the penalty for coefficients that are smaller. This adaptive behavior helps in improving the accuracy of coefficient estimates, making it more effective in certain scenarios compared to the regular Lasso estimator. The adaptive Lasso estimators are obtained as:

$$\Theta_{\text{B-ALasso}}(\lambda) = \text{argmin}(\Theta - L(R_{i,1}^{(j)}, V_{i,1}^{(j,k)}, V_{i,2}^{(j,k)}; \Theta) + \lambda \sum_{j=1}^{m} w_j |\theta_j|). \tag{13}$$

where λ is the tuning parameter and $P(\Theta) = \sum_{j=1}^{m} w_j |\theta_j|$ is the penalty term.

4.5 Class Imbalance

Class imbalance in a dataset occurs when the distribution of samples across different classes differ (Japkowicz & Stephen 2002). This is often a problem since the smaller class is often of more interest and importance. In our case, the class of importance is the defaults, which indeed is a minority class in our dataset. Class imbalances can significantly impact the performance of models, as they can lead to biased predictions and reduced accuracy.

There are different aspects that affect the degree of model performance and how models are influenced by class imbalance. One aspect could be that a smaller training dataset would lead to a greater effect of class imbalances in classification models. To deal with these kind of class imbalances, there are several proposed strategies. Firstly, the minority class can either be oversampled or the majority class can be undersampled until the classes are approximately equally represented. The dataset would, therefore, be modified to create a more balanced distribution of classes. Alternatively, one can assign distinct costs to the classification errors. This involves attaching different penalties for each type of classification mistake. Lastly, one can bias the classification algorithm towards the minority class when training the model. This can be done by adjusting the class weights or modifying the loss function. The goal is to be able to learn what features are important for both classes, not only the majority class. Due to this, the performance of the minority class would be improved (Barandela et al. 2003).

4.6 Statistical measures of performance and interpretability

In this thesis we have chosen to focus on three performance measures; type I error and type II error, ROC AUC score, and PCC score. Although, we also use the KS statistic and the Brier score. In this section, the scores are presented and explained. The selection of appropriate scores to assess model performance heavily relies on previous literature.

In our case, we can expect two types of mistakes: type I error and type II error. Either the model can incorrectly assign an applicant who has not defaulted to the defaulted category (type I error), or it can incorrectly assign the applicant that has defaulted to the non-default category (type II error). Therefore, it is often of interest to show the predictions of the applicants in a confusion matrix and to illustrate the correctness of the categorical predictions. The confusion matrix indicates correct and incorrect predictions for applicants, while the correctly classified proportions (PCC) represent the fraction of accurately classified applicants within the dataset (Gareth et al. 2013). According to Lessmann et al. (2015), PCC requires discrete class predictions, which are obtained by comparing the probability of default to a threshold τ. In our case, the threshold is 0.5. All applicants with a probability under 0.5 are assigned a value of 0 (non-defaults), and 1 (defaults) otherwise. The probabilities that are over the threshold are assigned to the defaultable ('positive') class, whereas those under the threshold are assigned to the non-defaultable ('negative') class. Thus, the type I errors are the false positive class, and the type II error are the false negative class. A lower threshold leads to lower false positives, however, it increases the false negatives, and vice versa. (Gareth et al. 2013).

Figure 2: Illustration of a ROC curve

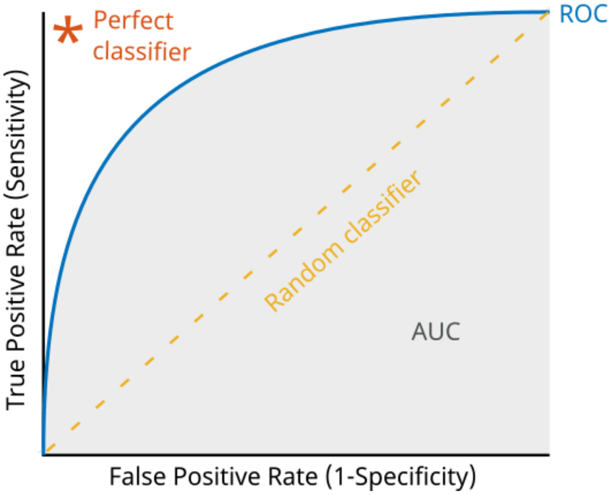


Figure 2 illustrates an example of a ROC curve. The perfect classifier is when it correctly assigns 'positive' class observations to the positive class and has a true positive rate of 1 for all thresholds. The random classifier returns random score values and has the same value for the true positive rate and the false positive rate of 0.5 for all thresholds. AUC is the area under the ROC curve.

A popular way to illustrate the relationship between the two types of errors for all possible thresholds is the Receiver Operating Characteristics (ROC) curve. Figure 2 illustrates an example of a ROC curve. The overall performance of the classification, summarized over the possible thresholds, is given by the area under the ROC curve (AUC). AUC assesses the discriminatory ability of the predictions. The ROC curve converging towards the left corner indicates a larger area under the curve (AUC), which corresponds to better classifiers. Comparing different classifiers using the ROC curve and AUC is useful because they consider all possible thresholds, offering a complete overview of performance (Gareth et al. 2013).

An alternative measure that assesses the correctness of the categorical predictions is the Kolmogorov-Smirnov (KS) statistic, which is also based on the probability of the outcome, but considers a fixed reference point. The fixed reference point serves as a benchmark for assessing the probability distribution of the outcome being measured. The KS statistic is the maximum difference between the cumulative score distributions of positive and negative classes (Lessmann et al. 2015).

A different performance measure that is used in credit scoring studies is the Brier score (BS), which is defined as the mean-squared error between the probability of x and the target binary response variable. BS is a global assessment, similarly to the AUC, since it considers the whole score distribution (Lessmann et al. 2015). For all the other mentioned measures (KS-statistic, BS, AUC and PCC) higher score reflects better performance of the model. However, in this case, a lower value of BS represents a better model (Dumitrescu et al. 2022).

4.6.1 Calculating error rates

In order to calculate credit losses and opportunity costs we first need to calculate the error rates. The false positive rate represents the proportion of negative instances (true negatives) that are incorrectly classified as positive (false positives) out of the total number of actual negative instances. In our case, this is when non-defaulted applicants are predicted to default. The type I error rate, also known as the false positive rate, is calculated with the following formula:

$$\mathrm{FP} = \frac{\sum \text{False Positives}}{\sum (\text{False Positives} + \text{True Negatives})} \tag{14}$$

The false negative rate represents the proportion of positive instances (true positives) that are incorrectly classified as negative (false negatives) out of the total number of actual positive instances. In our case this is when defaulted applicants are predicted to not default. This is also known as the type II error rate and it is calculated with the following formula:

$$\mathrm{FP} = \frac{\sum \text{False Negatives}}{\sum (\text{False Negatives} + \text{True Negatives})} \tag{15}$$

4.7 Economic measures of performance

In addition to examining the models' performance for each weight, we also want to evaluate the economic impact. As the bank's bottom line, we define the credit losses as the predicted non-defaults that actually do default (type II error) and the opportunity costs as the predicted defaults that do not default (type I error). The losses and revenues are calculated in order to evaluate the trade-off between the opportunity costs and credit losses for each model and class weight.

The assumptions when calculating the revenues are that the loan would only be paid at maturity and that it does not default. The calculated revenue is the sum of all interest payments during a loan's lifetime. Firstly, we assume monthly payments (PMT), which are calculated using e.(16). The amortization is calculated by multiplying the monthly interest rate (r) with the present value of the loan (PV). The interest payments are calculated using the monthly interest rates and the duration of the loans in months (n).

$$\text{PMT} = \frac{r \cdot PV}{1 - (1 + r)^{-n}} \tag{16}$$

The monthly payment is then multiplied by the duration of the loan in order to get the total payments for the lifetime of the loan. Lastly, the loan amount is subtracted in order to retrieve only the interest payments and not the amortization.

$$\text{Total Interest} = (\text{PMT} \cdot n) - \text{PV} \tag{17}$$

Losses are calculated as the sum of the defaulted loans and principal future total interest payments since we both lose the loan amount and the revenue from the loan. In addition, we also assume that the recovery rate is zero.[14]

5 Empirical strategy

5.1 Predicting the probability of default

The probability of default is estimated for three different dependent variables: total defaults within the whole dataset, total defaults within 12 months, and total defaults within 5 months. This is done in order to identify if the effect of the variables differs depending on the time before the loan defaults. In particular, we want to assess if there are features that define straight rollers, i.e., loans that default after five months with no payment, in comparison to a loan that defaults where a payment has been made. For each case of defaults, we firstly apply each model to the training subsample in order to evaluate if the model is accurately fitted, and secondly validate in order to review the performance. Lastly, the models are assessed on the test subsample to obtain the performance of each model.

[14] This is usually not true in practice, but scoring models should not take the recovery rate into account.

5.1.1 Tuning hyperparameters

As mentioned, hyperparameters are tuned in order to increase the performance of the models. In this thesis, we use the random search approach to achieve the tuning of the hyperparameters for each model. The tuning is done separately for total defaults, 12-month defaults, and 5-month defaults.

The parameters are similar for decision trees and the random forest. The hyperparameters tuned in this thesis are the most common ones used for decision trees: the maximum debt of the tree, the maximum number of leaves, the quality of the split, the number of features to consider when looking for the best split, parameters that consider the minimum number of samples required to split in an internal node, and the minimum number of samples required to be in a leaf node. The random forest is tuned with the same hyperparameters as mentioned, with the exception of the maximum number of leaves and the addition of the number of trees in the forest. The tuning is done when training the dataset with a five-fold cross-validation until we reach satisfactory results. If the results differ substantially from the training and validation, this is an indication of overfitting. Thus, the next step is to tune the hyperparameters such that the trade-off between overfitting and performance from the cross-validation is maximized. When this is achieved, the final step is to apply the tuned model to the test subsample.

The PLTR and the logistic regression have the same parameters that can be tuned. The hyperparameters that are tuned for the PLTR is the penalty term and an algorithm that is used in the optimization. In addition, we tune a logistic regression with a penalty term in order to obtain the penalized logistic regression.

5.1.2 Class weighting

An issue encountered within our datasets is class imbalance, where the proportion of the defaults is significantly lower compared to non-defaults. Specifically, the total defaults contain roughly 18% defaults, the defaults within 12 months contain 6% defaults and the defaults within 5 months contain merely 1.4% defaults. Consequently, our models tend to mainly predict non-defaultable applicants due to the skewed distribution of the classes. In order to address this, we have used different class weights that influence the models to predict more defaults. Thus, we put higher weight on the minority class in order to see whether the performance of our models improves. Tuning the class weights is beneficial since we get a better overview of how type I and type II errors differ as the weight for the minority class changes.

For each model, with the same hyperparameters designed in the previous section, we are testing different class weights on the training subsample in order to pick the class weights that we consider to be most suitable. The class weights can be presented as a percentage that reflects the importance of defaults within the dataset. A higher weight percentage indicates that the minority class is considered more important, while a lower weight percentage indicates less importance. All weights have the same percentage weight for the total defaults, defaults within 12 months, and defaults within 5 months. The resulting weight percentages for the minority class in this thesis are; balanced (both classes are of equal importance), 45%, 40%, 35%, and 20%. Lastly, we compare the results from these weights with results obtained without any class weights.

In the PLTR model, both the decision tree and the regression require the application of class weighting. Initially, the previously stated weights are applied to the one-split and two-step trees, resulting in distinct dummy variables for each weight. This is in order to capture bivariate threshold effects for each weight. Finally, we apply the same weights used in the decision trees to a penalized logistic regression, where the regression is performed for each weight-specific dataset.

6 Results

The results are divided into three sections: Section 6.1, the total defaults; Section 6.2, defaults occurring within 12 months; and Section 6.3, defaults occurring within 5 months. Each section incorporates the importance of variables, the ROC curves, the accuracy scores, and the error rates for each model. Furthermore, the accuracy scores and the error rates are also presented for chosen class weights. We then present the opportunity costs and interest losses that each model predicts for each class weight. In addition, we compare the predictions from the models to the outcomes in Section 6.4.

6.1 Total Defaults

6.1.1 Importance of Variables

The tree-based models have the ability to rank features by importance when predicting the probability of default. Accordingly, Figure 3 shows the ten most important variables for the random forest model, where the importance is based on how much including that variable improves the model's ability to accurately predict the probability of default. This is also known as the Gini importance and is normalized so that it is comparable across different variables. The variables that our models assume to be most important may vary across the models. However, we find that the tree-based models have similar results when it comes to which variables are considered the most important in the total defaults dataset.

Figure 3: Importance of variables predicting total defaults

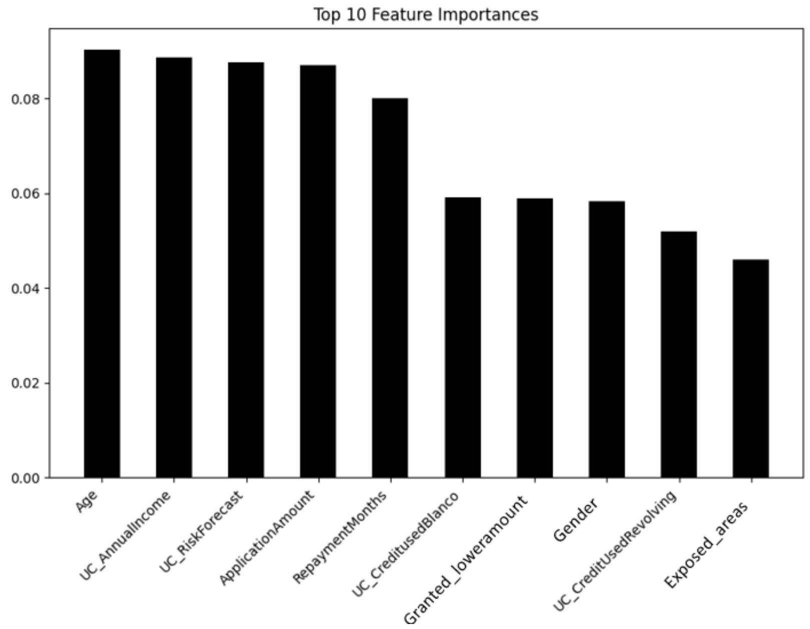
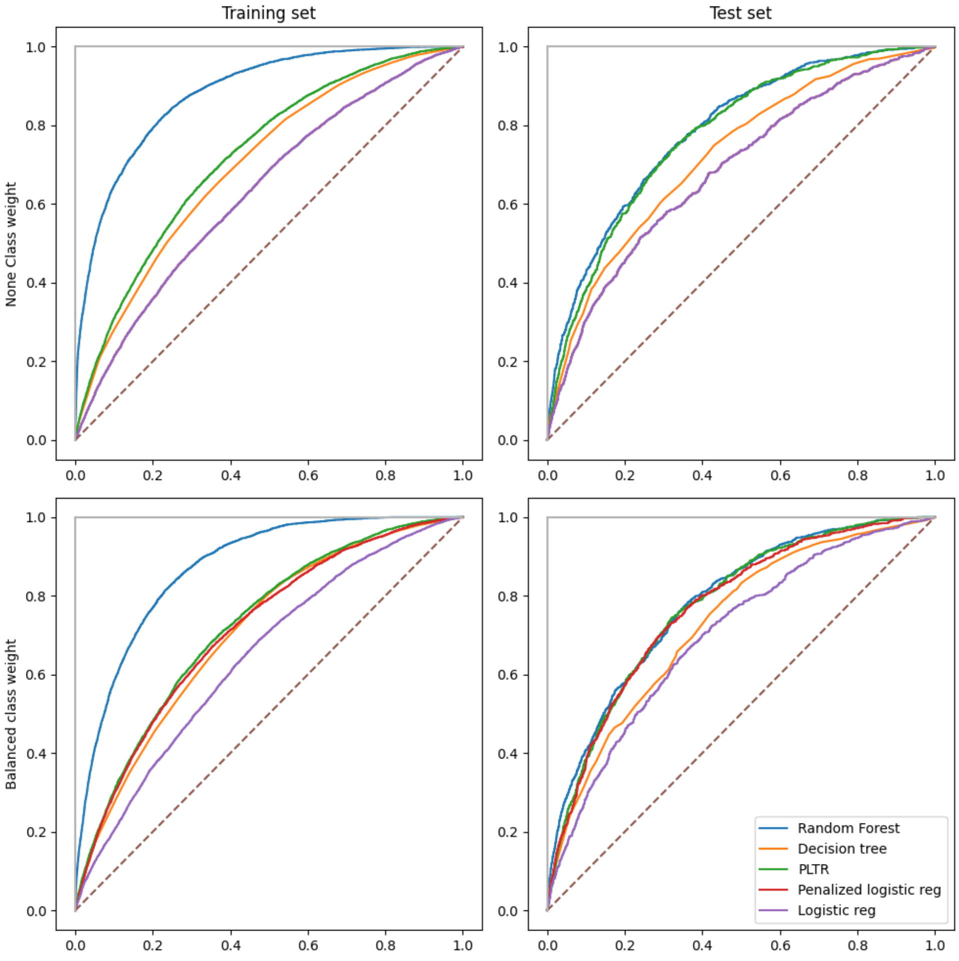


Figure 3 displays the top 10 features that the random forest model considers important when predicting defaults. The importance scores are based on the Gini importance which is normalized and sum up to 100%. In our dataset, There are a total of 30 variables. The feature importance score for a given feature is calculated as the reduction in the impurity of the tree when that feature is used for splitting the data. In this figure, "UC_Creditusedblanco" is the current debt that is not collateralized by the applicant at the disbursement date, and "Granted_loweramount" is whether the applicant has been granted a lower amount than what they applied for.

As displayed in Figure 3, age, annual taxable income, UC risk forecast measure, and application amount are the most important variables when predicting the total probability of default, with an importance score of almost 0.09 each. Usually, a lower age and a smaller application amount lead to a higher default risk. In addition, the duration of the loan is of large importance, although, the duration has a high correlation with the application amount. Providing a lower loan amount than requested by the customer increases the default probability. If the customer has an existing blanco debt (debt that is not collateralized) or a signed credit card agreement when applying for the loan, a higher default risk is expected. Lastly, the applicant's gender and where they live are important features as well. If the applicant lives in an exposed area, their default risk is expected to be higher.

6.1.2 Performance results

Figure 4: ROC curve results for total defaults



The ROC curves depicted in 4 compare the ROC curves of using a balanced class weight in the dataset with the ROC curves of using no class weights for the total defaults. The top row displays the results obtained by predicting defaults on both the training and test sets without any class weight. In contrast, the bottom row exhibits the results achieved by predicting defaults on both the training and test sets with a balanced class weight. Each line in the graph is represented by a model, for example, the blue line represents the ROC curve from the random forest.

Figure 4 shows four different ROC curves for the total defaults. we compare the resulting ROC curves without using any class weight to the ROC curves obtained when using a balanced class weight (weight percentage of 50%). The results from the training set are displayed on the left-hand side, whereas the results from the test set are shown on the right-hand side. Considering the information in Figure 4, there is no significant difference when using no class weight compared to using a balanced class weight for the tree-based models, which is an indication that none of these models are sensitive to class imbalances. The models that are based on the logistic regression do show small improvements, indicating that these models are more sensitive to imbalances in datasets. In addition, in all cases, there are noticeable differences between the performance in the training set and the test set. This is especially true for the random forest, which has a drastic decrease in performance out-of-sample. The Random forest model shows a clear indication of overfitting.

To examine the performance in more detail, we present four different accuracy measures for each model without class weights and with balanced class weights in Table 5. The model that has the best performance according to all scores, except the KS statistic, is the random forest. However, we can again see that random forest differs the most between the test and training datasets.

Table 5: Accuracy measures

(a) No class weights

Training set Results					Test set Results				
Model	PCC	ROC AUC	BS	KS statistic	Model	PCC	ROC AUC	BS	KS statistic
Random Forest	81.66%	88.22%	11.86%	18.00%	Random Forest	89.24%	78.50%	9.10%	10.02%
Decision Tree	80.14%	70.08%	14.67%	18.45%	Decision Tree	88.92%	72.21%	9.68%	9.49%
Log Regression	79.92%	63.29%	15.55%	19.84%	Log Regression	89.01%	68.27%	9.96%	10.94%
Penalized LR	79.92%	63.29%	15.55%	79.94%	Penalized LR	89.01%	68.27%	9.97%	10.94%
PLTR	80.17%	72.35%	14.40%	16.79%	PLTR	88.11%	77.44%	9.89%	6.01%

(b) Balanced class weight

Training set Results					Test set Results				
Model	PCC	ROC AUC	BS	KS statistic	Model	PCC	ROC AUC	BS	KS statistic
Random Forest	78.89%	87.18%	16.16%	12.41%	Random Forest	77.87%	78.13%	16.01%	12.83%
Decision Tree	57.39%	70.77%	21.90%	34.02%	Decision Tree	59.88%	72.90%	20.48%	34.74%
Log Regression	63.37%	64.47%	23.21%	17.97%	Log Regression	71.78%	69.95%	20.99%	17.95%
Penalized LR	65.30%	71.40%	21.68%	79.94%	Penalized LR	69.67%	76.84%	19.82%	24.03%
PLTR	65.67%	72.39%	21.37%	21.50%	PLTR	63.98%	77.36%	22.55%	31.26%

Table 5 show the PCC, ROC AUC, BS and the KS statistics for the different models. The left side of the table represent the performance results from the training dataset, while the left side show the performance result for the test dataset. Table (a) represent the accuracy measures when using no class weight, whereas the table (b) shows the results when using a balanced class weight.

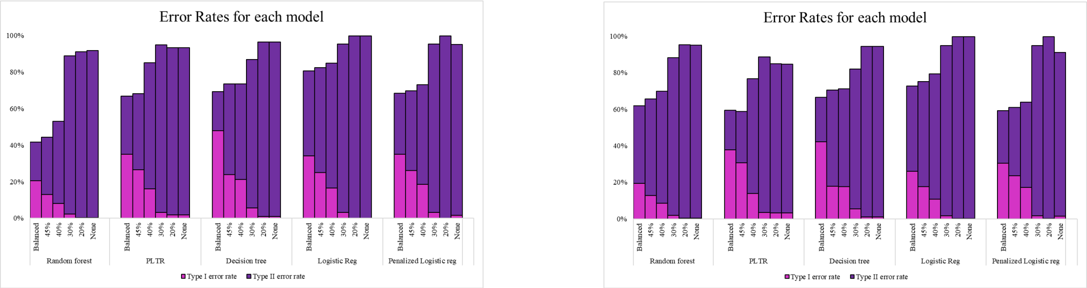
Looking at all the scores in general, both the PCC measure and the ROC AUC score have increased when predicting the test data. Especially the PCC score, since it has increased by approximately 10 percentage points. This could be due to time variation since our samples are sorted according to time. In addition, Collector Bank has changed its credit policy between the period of training and test data, with a restriction of the cut-off for granting a loan, which is expected to increase the variation further between the datasets. Therefore, the samples in the training and test data are likely to have different characteristics. In our case, this explains the increased accuracy in the test set: there are fewer defaults due to a lower cut-off strategy, resulting in higher class imbalance and increased PCC score because more non-defaults are classified correctly.

When looking more closely at the ROC AUC score, both for no class weights and balanced class weights in table 5, random forest has the highest score followed by the Decision tree and PLTR. These models are the best at minimizing the risk of false positives and false negatives. In addition, the penalized logistic regression shows similar results to the more complicated PLTR model when using a balanced class weight. Although, the PLTR model has a higher ROC AUC score with a difference of approximately 0.5 percentage points on the test set. Focusing on the PCC score, the random forest has the highest value followed by the logistic regression. Although the logistic regression presents the lowest ROC AUC score, the high PCC score can rather be a result of predicting mainly non-defaults due to the model being sensitive to class imbalance. Comparing the ROC AUC scores of all the models, it is observed that the more advanced models generally have higher scores, while the models that are easier to interpret tend to have lower scores.

6.1.3 Class weight results

Figures 5a and 5b show the percentage of the type I and type II errors for each model at each class weight. Figure 5a shows the results for the training set, whereas Figure 5b shows the resulting rates when predicting on the test set. The concept of Type I and Type II errors can be approached from two different perspectives. Type I error arises when the model incorrectly classifies non-defaulted applicants as defaulted. Thus, this error can be described as the opportunity cost of the bank, as it represents the potential revenue losses from rejected loan applications. Conversely, type II error refers to defaulted applicants that are incorrectly classified as non-defaulted. This error can therefore be seen as the bank's risk aversion as it depends on the accuracy with which the bank seeks to predict defaults. This means that higher prediction accuracy translates into a greater risk aversion by the bank. By comparing the error rates for each class weight and model, one can observe which model the bank should use based on risk aversion.

Figure 5: Resulting error rates predicting total defaults



(a) training set

(b) test set

Graphs 5a and 5b shows the error rates for each model and each class weight for the total defaults. The balanced weight is when the different classes are assigned the same importance and the 'none' represent no class weight. The 45% weight percentage is when the minority class is assigned 45% importance, the 40% weight percentage assigns 40% importance to the minority class, etc. The pink area is the type I error rate and the purple area represent the type II error rate. The sum of these areas represents the total errors for each model and weight. The resulting error rates are displayed for both the training set and the test set.

In general, there is a large difference between the type I and type II errors when using different class weights for each model. There is a distinct pattern in both graphs: not accounting for class imbalance results in more type II errors and fewer type I errors. Most of the models exhibit the lowest error rate when using a balanced class weight. At this class weight, the credit losses are minimized, but the opportunity costs are at their highest. It can be seen that the random forest has the lowest error rates on the training set, followed by PLTR and penalized logistic regression. Although, when looking at the test set, PLTR now shows lower error rates than the random forest. In addition, for the higher class weights penalized logistic regression has lower (or similar) error rates than PLTR.

In order to investigate the results from the class weights more thoroughly in terms of performance, we present Table 6. This table shows the number of observations that falls into the type I and type II error respectively, with the addition of the PCC accuracy and AUC score.

Table 6: Class weights performances on total defaults

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	5526	1424	79.29%	87.71%	Balanced	1443	393	78.12%	78.17%
45%	3494	2106	83.32%	88.33%	45%	1443	488	82.77%	78.15%
40%	2180	3027	84.49%	87.44%	40%	632	566	85.72%	78.40%
30%	599	5836	80.83%	76.27%	30%	139	796	88.86%	77.48%
20%	54	6119	81.61%	88.17%	20%	30	876	89.20%	78.26%
None	58	6155	81.49%	87.48%	None	27	874	89.26%	78.34%

(a) Random Forest

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	12859	1441	57.39%	70.77%	Balanced	3141	226	59.88%	72.90%
45%	6383	3340	71.03%	69.98%	45%	3141	485	78.35%	72.57%
40%	5673	3524	72.60%	70.80%	40%	1316	495	78.42%	72.21%
30%	1501	5474	79.22%	70.10%	30%	410	706	86.70%	72.47%
20%	237	6428	80.14%	70.08%	20%	67	863	88.92%	72.21%
None	237	6428	80.14%	70.08%	None	67	863	88.92%	72.21%

(b) Decision Tree

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	9369	2154	65.67%	72.39%	Balanced	2823	200	63.98%	77.36%
45%	7133	2805	70.39%	72.39%	45%	2823	259	69.69%	77.40%
40%	4330	4642	73.27%	64.66%	40%	1027	581	80.84%	70.58%
30%	846	6177	79.08%	64.37%	30%	251	787	87.63%	70.48%
20%	503	6154	80.17%	72.35%	20%	246	752	88.11%	77.45%
None	510	6146	80.17%	72.35%	None	247	751	88.11%	77.44%

(c) PLTR

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	9412	2239	65.29%	71.40%	Balanced	2280	264	69.69%	76.84%
45%	7019	2931	70.36%	71.40%	45%	2280	346	74.89%	76.87%
40%	4998	3660	74.20%	71.40%	40%	1284	430	79.58%	76.90%
30%	824	6211	79.04%	64.18%	30%	125	860	88.26%	69.48%
20%	39	6700	79.92%	63.28%	20%	2	920	89.01%	68.26%
None	425	6298	79.97%	71.37%	None	105	827	88.89%	76.95%

(d) Penalized Logistic Regression

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	9164	3131	63.37%	64.47%	Balanced	1925	436	71.87%	69.72%
45%	6680	3866	68.58%	64.58%	45%	1925	543	78.24%	69.91%
40%	4435	4599	73.08%	64.56%	40%	740	645	83.50%	69.87%
30%	824	6212	79.04%	64.18%	30%	99	874	88.41%	69.29%
20%	39	6700	79.92%	63.28%	20%	1	920	89.03%	68.00%
None	41	6700	79.92%	63.29%	None	1	920	89.03%	68.15%

(e) Logistic Regression

Table 6 represent the performance measures and the number of observations that falls into the different error types for each model and class weight. CW stands for the different class weights, Type I and type II errors represent the number of observations that falls into each error. On the left-hand side, we have the results from predicting on the training set, whereas, on the right-hand side, we have the results from predicting on the test set.

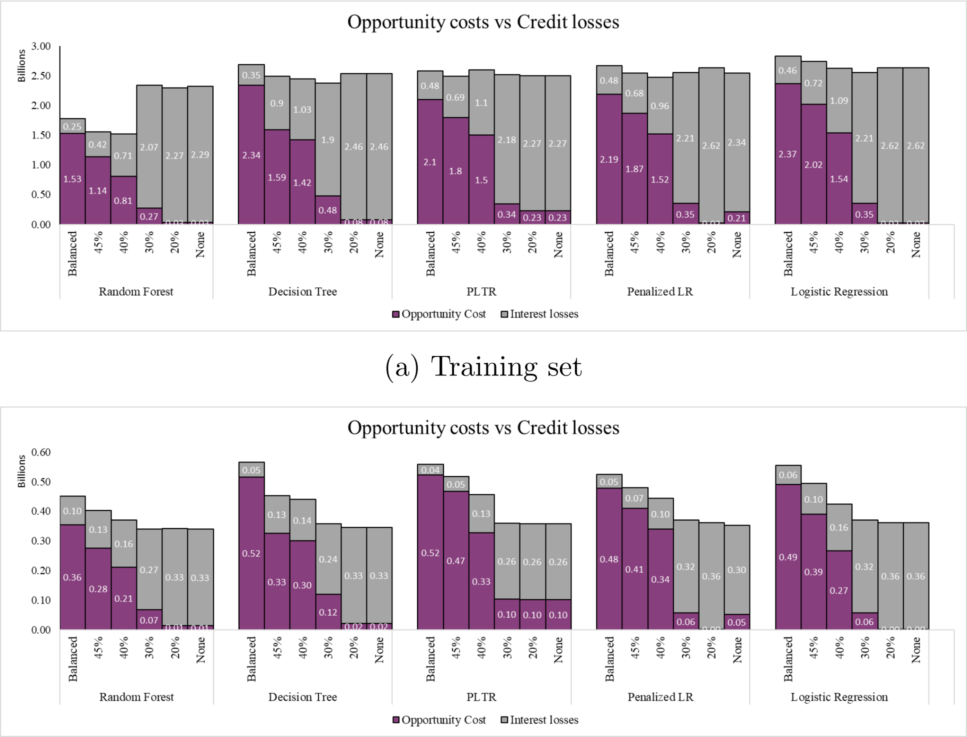
As in earlier cases, Table 6 shows significant differences in the results obtained when predicting on the training set and the test set. Again, the random forest model displays a noticeable decrease in ROC AUC scores. In contrast, linear models such as the penalized logistic regression exhibit a distinct increase in performance scores when predicting on the test set. Otherwise, Table 6 shows that the number of observations that fall into type I errors increases, whereas the number of observations in type II errors decreases as the class weights increase.

Moreover, in accordance with the decreased amount of type I errors due to lower class weights, all the models show an improved accuracy rate. This could be due to the model focusing less on predicting false negatives, and in turn, predicting more non-defaults and type II errors. However, no clear pattern is observed for the ROC AUC score for the different weights, particularly for the random forest and decision tree models. The scores remain similar regardless of the class weight used. Whereas, we see that the regular logistic regression model exhibits the most significant improvement in the ROC AUC score among the models when using class weights. This strengthens the case for more advanced models being able to handle large class imbalances better than simpler models and suggests a trade-off between interpretability and the ability to handle advanced issues.

6.1.4 Opportunity costs vs credit losses

As previously discussed, the type I errors can be thought of as the opportunity cost for the bank: these are the non-defaulting applicants that are incorrectly predicted as defaulted. We define the opportunity cost as the revenue losses in terms of paid interest if the loan were granted. In contrast, the type II errors portray the credit losses that the bank carries, since these are the defaulted applicants that are wrongly classified as non-defaults. We have specified the credit losses as the interest made based on the granted loan and the amount of the loan, without any recovery rate. Figure 6 displays the credit losses and the opportunity costs that the bank accounts for each model and class weight. The aim is to find the weight that gives the bank minimum total losses, which is the sum of credit losses and opportunity costs.

Figure 6: Opportunity Costs vs Credit Losses predicting total defaults



(a) Training set

(b) Test set

Figure 6 displays the opportunity costs and credit losses in terms of interest and loan size, for both the training set and test set. The x-axis displays the class weights for each model, and the y-axis represents the total losses in billion SEK. The opportunity cost is represented by the purple field and the credit losses are represented by the grey field. In both of the samples, each model has resulting opportunity cost and losses for each class weight.

As shown in figure 6, the credit losses are maximized, and the opportunity costs are minimized when no class weights are used. Whereas the balanced class weight shows the lowest credit losses and highest opportunity costs among the provided weights. However, the overall results between the training set and test set differ significantly, which could be explained by the different sample sizes or the sample variation. The credit losses are expected to be higher in the training set compared to those in the test set due to the aforementioned change in the bank's cut-off strategy. The class weight that minimizes total losses in the training set varies across the models. Among them, the random forest consistently achieves the lowest losses, with minimum losses at a weight percentage of 40%. In the test set, the models exhibit more similar results. The class weights that achieve the lowest total losses are either 20% or no class weight. However, using these weights leads to maximized credit losses, as the models tend to predict more non-defaultable applicants.

6.2 12 months defaults

6.2.1 Importance of variables

Figure 7: Importance of Variables predicting 12m defaults

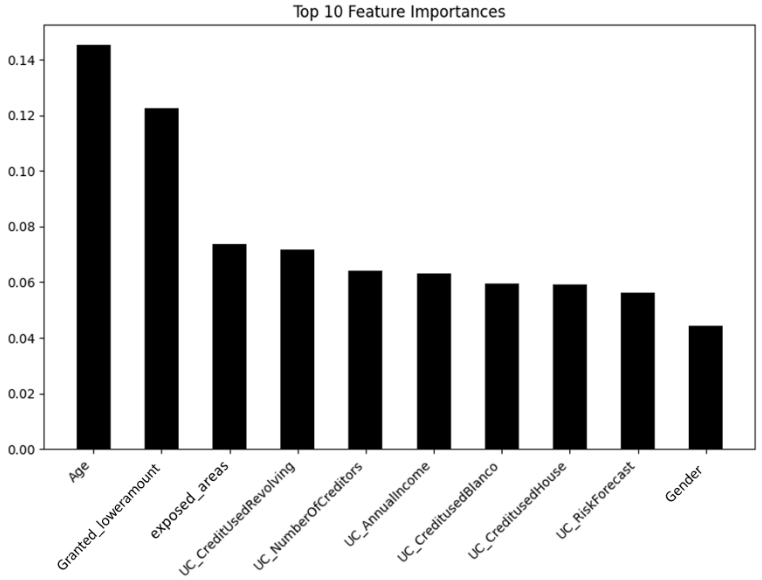


Figure 7 displays the top 10 features that the random forest model considers important when predicting 12 month defaults. The importance scores are based on the Gini importance which is normalized and sum up to 100%. In our dataset, there are a total of 30 variables. The feature importance score for a given feature is calculated as the reduction in the impurity of the tree when that feature is used for splitting the data.

In order to investigate if the most important variables differ between 12 months and total default, we present Figure 7. It shows the top 10 important features for the random forest when predicting the 12-month probability of default. Compared to total defaults, the new variables of importance for defaults within 12 months are the number of creditors, which indicates how many loans the applicant has and whether the customer has a mortgage (UC_creditusedhouse). If the applicant has a lower number of creditors and a mortgage it would result in a lower default risk. In addition, the variable that is not of importance anymore is the application amount. Instead, if the applicant has been granted a lower amount than what they applied for is of more importance. Also, it is noticeable that living in an exposed area has greater importance for the 12-month default.

6.2.2 Performance of models for different class weights

We provide similar figures and tables as in the earlier case in order to analyze whether the results differ between the total defaults and the results obtained from predicting the defaults within 12 months. Thus, Figure 8 shows four different ROC curves for the defaults within 12 months, which compares the ROC curves without any class weights to the ROC curves obtained when using the balanced class weight.

Figure 8: ROC curves 12 month defaults

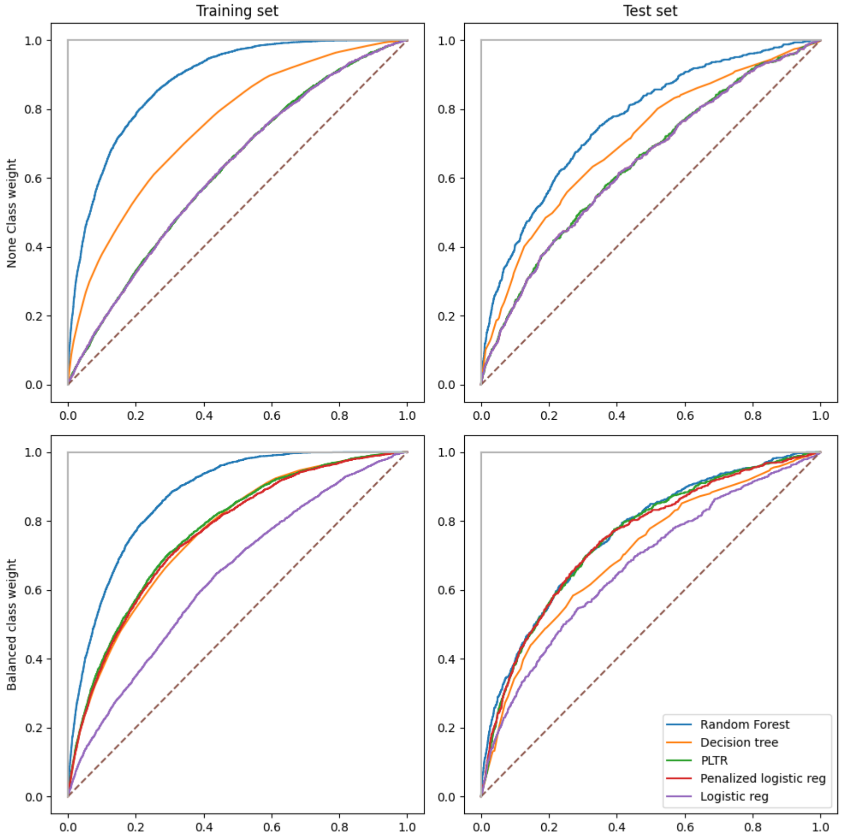
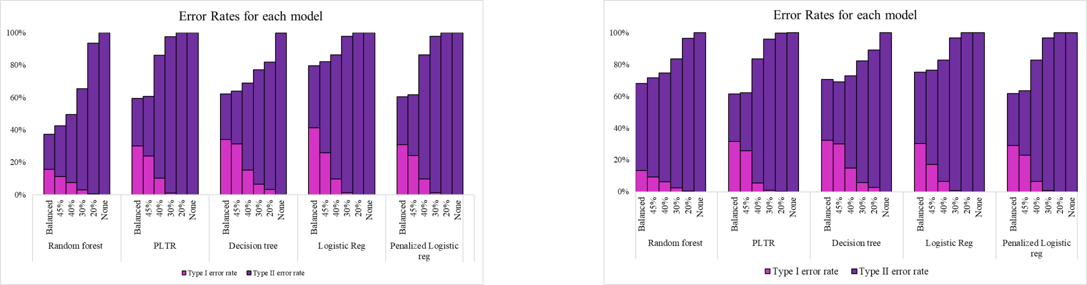


Figure 8 represent the ROC curves for the 12-month defaults for the different models on the training and test dataset. The top row of ROC curves displays the results obtained by predicting defaults on both the training and test sets without any class weight. In contrast, the bottom row exhibits the results achieved by predicting defaults on both the training and test sets with a balanced class weight.

In comparison to the previous section, there are more distinct differences for the 12-month defaults, even if we observe similar patterns. We see a clear difference between the ROC curves produced on the training set when using balanced class weights. The simpler models, such as the penalized and non-penalized logistic regressions, show a significant improvement in performance. In contrast, the machine learning models have similar performance in the training set regardless of class weight choice. Whereas, the random forest model overfits with lower performance in the test set. Although, the performance of the decision tree differs between the training and the test set. Unlike in the case of total defaults, all machine learning models show signs of overfitting for the 12-month dataset. The reason for this could be that modeling total defaults may provide a broader and more representative view of the data, making it more difficult for the model to overfit on specific patterns or noise in the data.

Figure 9: Resulting error rates for 12m defaults



(a) training set

(b) test set

Figure 9 shows the error rates for the different models and each class weight for the 12-month defaults. Panel A shows the resulting error rates obtained when predicting on the training set and panel B shows the resulting error rates from the test set. The balanced weight is when the different classes are assigned the same importance and the 'none' represent no class weight. The pink area is the type 1 error rate and the purple area represent the type 2 error rate. The sum of these areas represents the total errors for each model and weight.

Figures 9a and 9b shows the error rates for the different models and class weights. Without any class weights, all models have a type II error rate of 100%. The indication of this is that the models predict that all applicants are non-defaulting. This underscores the importance of choosing the right weighting scheme. Similarly to the total defaults, when the class weights are tuned, there is a distinct pattern that the type II error rates decrease and the type I error rates increase. The risk for the bank becomes lower when increasing the class weights, but the opportunity costs for the bank simultaneously increase. Similarly, the class weights can, therefore, be illustrated as the amount of risk the bank is willing to take.

The overall results indicate that the use of the 'balanced' class weights generally leads to the lowest total error rates across most models. In the training set, the random forest produces the lowest total error rate, while the logistic regression has the highest total error rate, both using the balanced class weight. However, the results from the test set are noticeably different, as the penalized logistic regression and PLTR produce the lowest total error rate at 60% for the balanced class weight. However, the error rates for all models are more similar, with the other error rates in the 60%-70% range.

More detailed results from predicting defaults within 12 months on both samples are presented in Table 7. For each model and sample, we present the number of observations that fall into the type I error and type II error. Additionally, we present the PCC accuracy measure and the AUC score for two class weights. As a common pattern among all models and as previously seen for the total defaults, the number of observations that fall into type I error increases as we weigh the defaults higher than the non-defaults. At the same time, the number of observations within the type II error decreases. In comparison to the total defaults, it is more evident for the 12-month defaults that the more advanced methods, such as random forests and decision trees, are almost unaffected by the choice of class weights. At the same time, all of the methods that use logistic regression, including the PLTR, show a significantly higher ROC AUC score when using larger weights for defaults. Given the ROC AUC scores, the random forest model still performs the best in general. However, when looking only at the balanced weights, the random forest, PLTR, and the penalized logistic regression all perform quite similarly. Additionally, it is again observed that the PCC score among the models increases in the test set, which is mostly due to the aforementioned change in the bank's cut-off strategy.

Table 7: Class weights performances on 12m defaults

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	7335	658	83.99%	90.03%	Balanced	1870	344	82.26%	75.57%
40%	3482	1274	90.47%	90.10%	40%	1025	429	88.35%	75.91%
None	0	3027	93.94%	91.14%	None	0	679	94.56%	76.15%

(a) Random Forest

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	15928	852	66.39%	76.09%	Balanced	3809	261	67.39%	70.66%
40%	7064	1627	82.59%	75.44%	40%	1753	394	82.80%	71.07%
None	0	3023	93.94%	75.13%	None	0	680	94.55%	70.60%

(b) Decision Tree

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	14144	883	69.90%	76.94%	Balanced	3809	261	67.39%	70.66%
40%	4739	2298	85.90%	64.97%	40%	1753	394	82.80%	71.07%
None	0	3027	93.94%	62.19%	None	0	680	94.55%	70.60%

(c) PLTR

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	14402	901	69.35%	76.13%	Balanced	3607	212	69.40%	74.76%
40%	4542	2318	86.26%	64.96%	40%	1215	466	86.53%	67.08%
None	0	3027	93.94%	62.11%	None	0	679	94.56%	63.84%

(d) Penalized Logistic Regression

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	19315	1162	58.98%	63.77%	Balanced	3579	305	68.88%	67.09%
40%	4543	2317	86.26%	64.96%	40%	752	519	89.82%	67.56%
None	0	3027	93.94%	62.11%	None	0	680	94.55%	64.07%

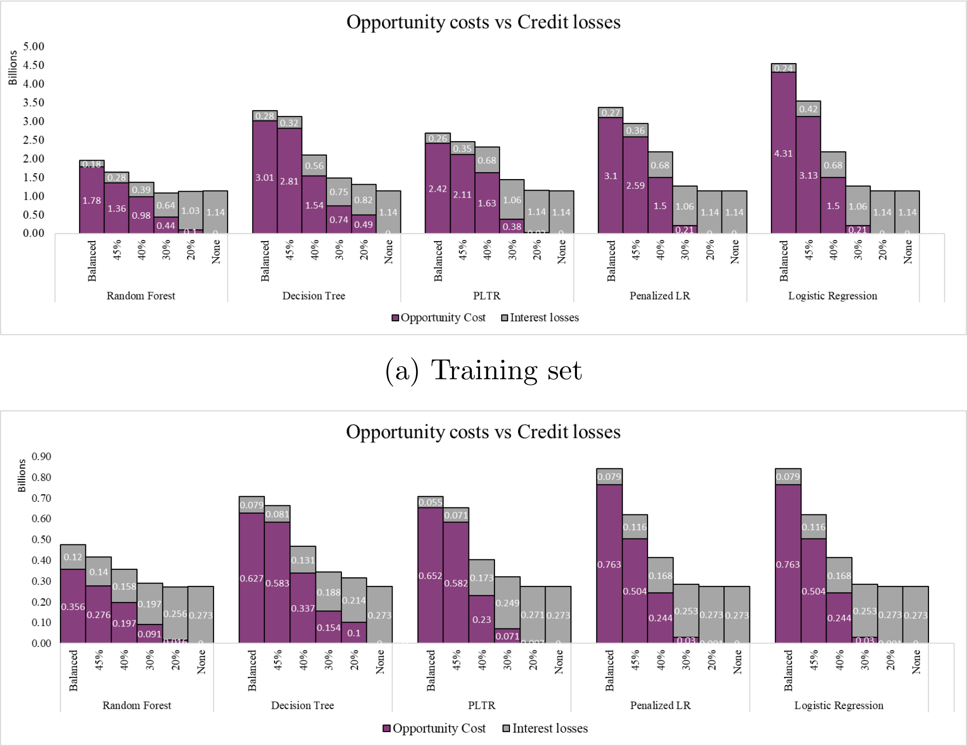
(e) Logistic Regression

Table 7 represent the performance measures and the error types for each model and class weight. CW stands for class weights; balanced, 40%, and no class weight. The type I and type II errors represent the number of observations that falls into each error. Where PCC and AUC scores are the performance measures. On the left-hand side, we have the results from predicting on the training set, whereas, on the right-hand side, we have the results from predicting on the test set.

6.2.3 Opportunity costs vs credit losses

As discussed before, type I and type II errors have an economic interpretability of opportunity costs and credit losses. The aim is to be able to choose the weight that produces the lowest amount of total losses, which is the sum of the opportunity cost and credit losses.

Figure 10: Opportunity Costs vs Interest Losses for 12m defaults



(a) Training set

(b) Test set

Figure 10 displays the resulting credit losses and opportunity costs for the defaults within 12 months. The opportunity cost is represented by the purple field and the losses are represented by the grey field. In both of the samples, each model has resulting opportunity cost and losses for each class weight. The y-axis represent the total amount of losses expressed in billion SEK.

The resulting opportunity costs and credit losses are displayed in Figure 10. In general, the 12-month default predictions exhibit lower credit losses in comparison to the total defaults. The reason behind this is that the models predict fewer type II errors, and instead, predict more type I errors resulting in higher opportunity costs. Since this is the case, the most beneficial class weights are either using no class weights, or the slightly tuned class weight percentage of 20%. When we do not use any class weight, all models predict only non-defaultable applicants, which leads to a greater loan portfolio but also greater credit losses.

6.3 5 months defaults

6.3.1 Importance of variables

When predicting the defaults within 5 months, other features could be of importance as they may be outstanding characteristics among these defaults in comparison to the earlier cases. Therefore, we present Figure 11 that contains the top 10 important features for the Random Forest when predicting the 5-month probability of default, also known as the straightrollers. In comparison to previous datasets, how many creditors the applicant has is the most important feature. If the applicant has many creditors, the risk of going directly to debt collection without making any payments increases. In addition, a new variable that has not been important in previous models is the debt ratio. This is defined as the ratio between the applicants' income and their total debt (including mortgage). A larger ratio leads to higher default risk within 5 months.

Figure 11: Importance of Variables for 5m defaults

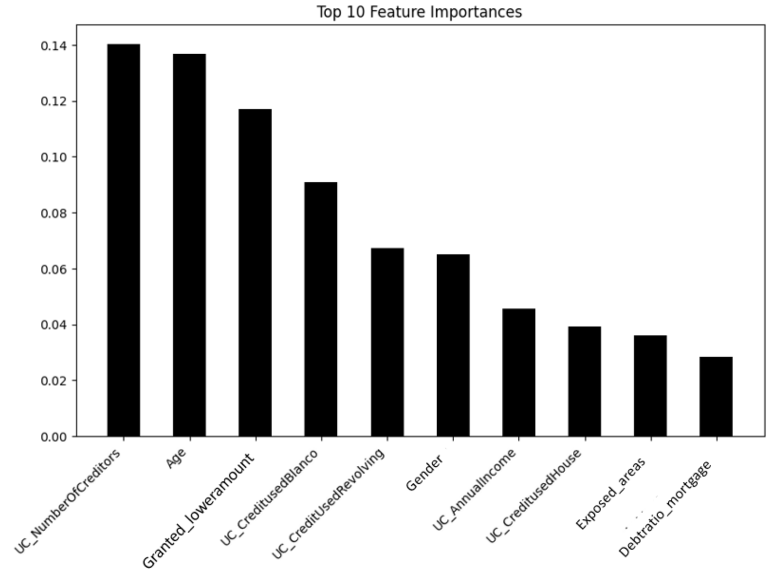


Figure 11 displays the top 10 features that the random forest model considers important when predicting 5 month defaults. The importance scores are based on the Gini importance which is normalized and sum up to 100%. In our dataset, there are a total of 30 variables. The feature importance score for a given feature is calculated as the reduction in the impurity of the tree when that feature is used for splitting the data.

6.3.2 Performance of models for different class weights

The same analysis as previous cases is done for the defaults within 5 months. Overall, the results from the defaults within 5 months are qualitatively similar to the results from the defaults within 12 months. Because of this, we discuss the overall results from the 5-month defaults compared to the 12-month defaults in this section and present the resulting figures and tables in Appendix 8.3.

The resulting ROC curves are similar to those observed in the 12-month defaults, but the shapes of the curves in the 5-month defaults are more distinct. The 5-month defaults show significant differences in the ROC curves between the training set and the test set for the Random forest and decision tree. Additionally, this difference is also shown in the PCC and AUC scores. This could be due to the 5-month defaults providing even a smaller view of the data, making it easier for the models to overfit on specific patterns and noise in the data. In general, the resulting ROC curves, PCC and AUC scores strengthen our previous results: the models based on logistic regression are more sensitive to class imbalance than the tree-based models. Random forest is still the best-performing model, whereas PLTR and penalized logistic regression produce slightly lower scores.

The error rates in general produce slightly lower error rates for the 5-month defaults in comparison to the 12-month defaults, especially on the test set. This result is intriguing and could depend on several factors. We conjecture that applicants that default within 5 months may have outstanding characteristics, such as a high number of loans, especially Blanco loans, and higher debt ratios, making them easier to detect. Another reason could also be the aforementioned change in the bank's cut-off strategy. The opportunity cost and the credit losses are also similar to the 12-month defaults. The results for the 5-month defaults may be more distinct, which could be due to the defaults within the dataset being considerably low. Consequently, using class weights increases the opportunity costs since it increasingly predicts false positives, and simultaneously minimizes the credit losses at a glacial pace. Because of this, the results suggest that the total losses are minimized when using no class weight.

6.4 Prediction vs outcome

In this section, we compare the predicted default probabilities and the actual default rates within 12 months. This is in order to compare the probability of default measures created by Collector Bank, given that their measure is based on defaults within 12 months. Building upon previous results, the predicted default probabilities are given by the Random Forest and penalized logistic regression. The underlying cause of this is that the random forest has favorable performance, meanwhile having less total loss in terms of opportunity costs and credit losses. Whereas the penalized logistic regression has decent performance while generating decent losses. Despite this, it is worth noting that the penalized logistic regression is easier to interpret and the random forest is more complex. Furthermore, we do not use any class weight for this analysis since this generates minimum losses in the 12-month default case.

Based on Figure 12, it is evident that the actual defaults fluctuate a lot during the whole time period, especially during unexpected economic events. In particular, fluctuations can be seen at the beginning of 2020, which was when the pandemic hit. During this period there is an approximate 2 percentage points increase in the default rate. Another noticeable increase of 3 percentage points is at the beginning of 2022, which could be mostly due to the increased inflation and higher interest rates. However, it is worth mentioning that active loans within the last 12 months were dropped, making the average higher due to fewer observations. Otherwise, the default rate in the dataset is consistently in the 4%-6% range.

Figure 12: Prediction vs Outcome for 12m defaults

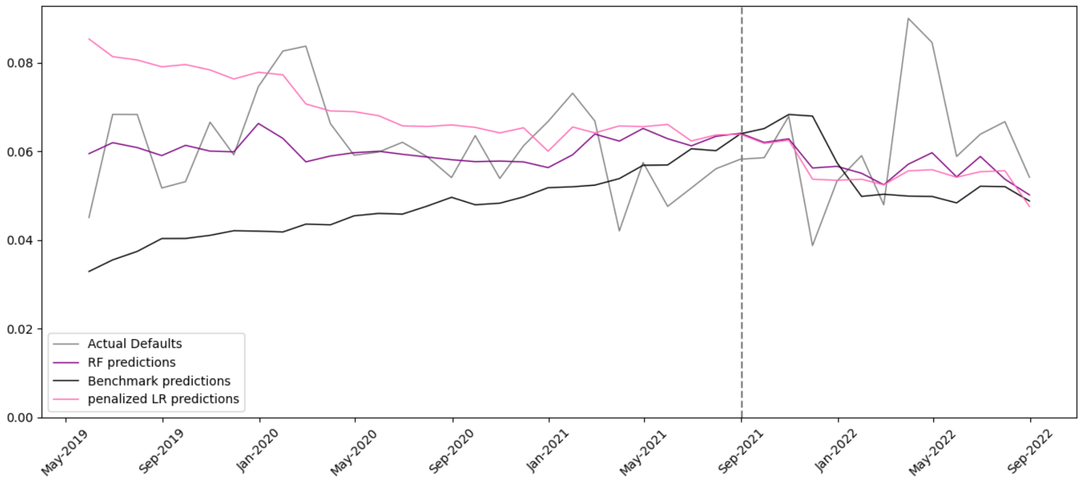


Figure 12 displays the average probability of default each month during the whole time period for all the customers. Benchmark predictions stand for the model currently used by Collector Bank for estimating the probability of default. RF predictions are the estimated probability for the random forest model and penalized LR predictions are the probabilities for the penalized logistic regression model, both without any class weight. Actual defaults are the percentage of customers that actually do default within 12 months. The dotted line represents the split into training and test samples.

Initially, the penalized logistic regression overestimates the default risk compared to the actual defaults. This might be because the actual defaults fluctuate, making the model predict higher default risk on average. As the defaults are less fluctuating, the overestimation diminishes over time and becomes more aligned with the actual defaults. However, the predictions on the test set differ from the predictions on the training set, as the model now underestimates the default risk. Therefore, it appears that the penalized logistic regression has difficulties to learn complex patterns in the training data, making the predictions on the test set less accurate.

The predictions on the training set made by the random forest differ substantially from the penalized logistic regression. Initially, the random forest predicts almost 2% lower default risk than the penalized logistic regression. The random forest has a smoother curve that follows the actual defaults, even if there is an indication of minor underestimations throughout the time period. However, both models have similar predictions on the test set. Even if the default spikes are hard to capture, the random forest captures these better than the penalized logistic regression. Hence, it is evident that the random forest acquires the characteristics of the defaultable applicants, making it easier to detect these during unexpected economic events.

Another interesting observation is that the benchmark underestimates the defaults risk for the majority of the time period, especially during the time period of the training sample. This is deviating compared to both the random forest and the penalized logistic regression. The penalized logistic regression almost mirrors the benchmark predictions, whereas the random forest is somewhere in between.

6.5 Discussion

In this section, the results are summarized and discussed. More specifically, the models' performance with regard to accuracy and interpretability is discussed. In addition, we discuss the models' reliability and sustainability in terms of regulation and ethics.

Starting with the performance of the models, the predictions gathered from the random forest have the best performance with regard to PCC and ROC-AUC scores. This applies to total defaults, 12-month defaults, and 5-month defaults. Without any class weight, the more advanced methods such as the random forest clearly outperform the more simple methods based on the logistic regression. This is in accordance with the studies done by Brown & Mues (2012), Moscato et al. (2021), and Trivedi (2020). In comparison to Dumitrescu et al. (2022), our findings suggest that the PLTR outperforms the regular logistic regression, but performs equally to the penalized logistic regression when predicting the total defaults, 12-month defaults, and the 5-month defaults. This evidence suggests that the dummies created from the one- and two-step decision trees have a minor effect on the performance of the PLTR model. Consequently, the penalized logistic regression compensates in terms of interpretability and performance, still performing better than the regular logistic regression. Although interpretability is an important factor, it comes as a cost of performance as the random forest still provides increased accuracy.

When adding the class weights a different conclusion emerges, especially when predicting defaults within 12 months and 5 months. The findings in the results suggest that the simpler models are more sensitive to class imbalance compared to the more advanced models. Accordingly, the resulting performance for the PLTR and the penalized logistic regression has drastically improved, especially when predicting defaults within 12 months and 5 months. The same improvement is not as evident when applying the weights to the total defaults, since the class imbalance is not equally comprehensive. Although the random forest still exhibits the best performance, it is worth noting that the PLTR and penalized logistic regression methods are not far behind in terms of their performance. Consequently, if both interpretability and performance are taken into account, the penalized logistic regression would be preferable. This can be explained by the fact that the random forest is more complex and harder to interpret, which may not compensate for slightly better performance. Even if the results from the PLTR when using class weights are more in line with Dumitrescu et al. (2022), the performance is still equal to the penalized logistic regression. Again, this suggests that the non-linear effects captured by the decision trees have a small impact.

Our analysis shows that most of the models perform the best in terms of total error rate when using a balanced class weight, with the random forest having the lowest error rate among all models. However, the use of class weights when predicting defaults also implies a trade-off between accuracy and loan approvals. The suggested results from using class weights imply that lower class imbalance would result in predicting the defaults more accurately, but at the same time denying more applicants that do not actually default (assuming the same cut-off strategy between different class imbalances). Hence, a higher weight percentage on the minority class would result in higher opportunity costs but lower credit losses. With the same cut-off as the earlier case, a greater class imbalance would make the bank grant more loans but it would increase the risk of defaults within the portfolio. Ultimately, the choice of class weighting should be considered based on the banks' risk aversion and not only the minimization of errors.

Lastly, when considering which model to use, the bank should also carefully consider what variables to include. For the 12-month defaults, exposed areas is the variable with the third most explanatory variable for the probability of default. If this variable is included in the model, the bank has to be aware of the Swedish Consumer Agency guidelines on fair credit scoring. The guidelines include that lenders should be aware of any biases against ethnic or socioeconomic groups and take steps to eliminate the bias. Even if including the exposed areas as a variable increases the models explanatory power, there is a risk of breaking the Consumer Agency Guidelines. In addition, if banks choose to use the random forest model that shows the best performance, they also need to be aware that there is a larger risk of bias when using more advanced methods. They also need to be able to present the more advanced models in an easy and interpretable way.

7 Conclusion

In this paper, we investigate and compare different credit scoring models, especially whether machine learning approaches outperform traditional models. We explore the recently proposed method called the PLTR model, which is a combination of machine learning and traditional logistic regression. Firstly, we compare the five models; random forest, decisions tree, PLTR, penalized logistic regression, and regular logistic regression, in terms of performance, such as accuracy measures and ROC curves, at different class weights. Secondly, we examine the economic interpretations of the type I error and type II error for each model at different class weights. Thirdly, we compare the predictions of the defaults within 12 months from the best-performing models to the benchmark prediction provided by the bank.

The main purpose of this paper was to identify the most effective and practical approach for credit scoring in the Swedish retail banking context. The findings suggest that the model that most accurately predicts the defaults is the random forest, with minimized error rates and costs. However, the random forest can be considered a "black box" due to its complexity, and gaining a full understanding of the models' decision process is almost infeasible, which could be a potential problem for Swedish Consumer Credit legislation and FSA guidelines. For this reason, the optimal substitute for the random forest is the penalized logistic regression according to our findings. This model compensates for interpretability due to its simplicity and transparency, but it comes with the cost of less accurate predictions.

Another aspect to be considered is the use of class weighting the model, although the more advanced models are proven to handle class imbalances better than the simpler models. The rationale behind this is the trade-off between the potential opportunity costs and credit losses that arises from incorrectly assigning applicants to a certain class. This trade-off is therefore based on the risk that the bank is willing to take. Ultimately, the banks' risk aversion should be taken into account when choosing a credit scoring model.

It is evident that the results obtained from the training and test samples have significant differences. We argue that the cause of this phenomenon is the modification of the bank's cut-off strategy between the two subsamples. A lower cut-off strategy leads to fewer defaults, which increases the class imbalance in the test sample. Consequently, these outcomes yield higher accuracy, as a greater number of non-default cases are correctly classified. Thus, an opportunity for future research would involve conducting an analysis regarding the optimal approach for data splitting. For instance, comparing and evaluating different ratios for data splitting in order to investigate the sample variation further.

References

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J. & Vanthienen, J. (2003), 'Benchmarking state-of-the-art classification algorithms for credit scoring', *The Journal of the Operational Research Society* **54**(6), 627–635.
URL: *http://www.jstor.org/stable/4101754*

Barandela, R., Sánchez, J. S., Garcıa, V. & Rangel, E. (2003), 'Strategies for learning in class imbalance problems', *Pattern Recognition* **36**(3), 849–851.

Barocas, S. & Selbst, A. D. (2016), 'Big data's disparate impact', *California Law Review* **104**(3), 671–732.
URL: *http://www.jstor.org/stable/24758720*

Bergstra, J. & Bengio, Y. (2012), 'Random search for hyper-parameter optimization', *Journal of Machine Learning Research* **13**(Feb), 281–305.
URL: *http://www.jmlr.org/papers/v13/bergstra12a.html*

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.
URL: *https://www.sciencedirect.com/science/article/pii/S095741741101342X*

Chen, D., Li, X. & Lai, F. (2017), 'Gender discrimination in online peer-to-peer credit lending: Evidence from a lending platform in china', *Electronic Commerce Research* **17**(4), 553–583.
URL: *https://doi.org/10.1007/s10660-016-9247-2*

Cigsar, B. & Deniz, U. (2018), 'The effect of gender and gender-dependent factors on the default risk', *Revista de Cercetare si Interventie Sociala* **63**, 28.

Commission, F. T. et al. (2019), 'Big data-a tool for inclusion or exclusion? understanding the issues (2019)'.

Dastile, X., Celik, T. & Potsane, M. (2020), 'Statistical and machine learning models in credit scoring: A systematic literature survey', *Applied Soft Computing* **91**, 106263.
URL: *https://www.sciencedirect.com/science/article/pii/S1568494620302039*

Duarte, J., Siegel, S. & Young, L. (2012), 'Trust and Credit: The Role of Appearance in Peer-to-peer Lending', *The Review of Financial Studies* **25**(8), 2455–2484.
URL: *https://doi.org/10.1093/rfs/hhs071*

Dumitrescu, E., Hué, S., Hurlin, C. & Tokpavi, S. (2022), 'Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects', *European Journal of Operational Research* **297**(3), 1178–1192.
URL: *https://www.sciencedirect.com/science/article/pii/S0377221721005695*

Finansinspektionen (2022), 'Rapport svenska konsumtionslån - fi.se'.
URL: *https://fi.se/contentassets/379333a8d53045a9ab7a7bfa9e276b36/svenska-konsumtionslan-2022n.pdf*

Gareth, J., Daniela, W., Trevor, H. & Robert, T. (2013), *An introduction to statistical learning: with applications in R*, Spinger.

Goddard, M. (2017), 'The eu general data protection regulation (gdpr): European regulation that has a global impact', *International Journal of Market Research* **59**(6), 703–705.

Greenberg, E., Goodman, L. & Zhu, J. (2019), 'Comparing credit profiles of american renters and owners', *Urban Institute* .
URL: *https://www.urban.org/sites/default/files/publication/78591/2000652-Comparing-Credit-Profiles-of-American-Renters-and-Owners.pdf*

Gutiérrez-Nieto, B., Serrano-Cinca, C. & Camón-Cala, J. (2016), 'A credit score system for socially responsible lending', *Journal of business ethics* **133**(4), 691–701.

Japkowicz, N. & Stephen, S. (2002), 'The class imbalance problem: A systematic study', *Intelligent data analysis* **6**(5), 429–449.

Konsumenternas (2022), 'Kreditprovning och kreditupplysning'.
URL: *https://www.konsumenternas.se/lan–betalningar/lan/sa-fungerar-ett-lan/kreditprovning-och-kreditupplysning/*

Kreditupplysningslag (1973:1173) (1973), Svensk författningssamling.
URL: *https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/kreditupplysningslag-19731173*$_sfs - 1973 - 1173$

Kruppa, J., Schwarz, A., Arminger, G. & Ziegler, A. (2013), 'Consumer credit risk: Individual probability estimates using machine learning', *Expert Systems with Applications* **40**(13), 5125–5131.
URL: *https://www.sciencedirect.com/science/article/pii/S0957417413001693*

Kuhn, M., Johnson, K. et al. (2013), *Applied predictive modeling*, Vol. 26, Springer.

Lessmann, S., Baesens, B., Seow, H.-V. & 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.
URL: *https://www.sciencedirect.com/science/article/pii/S0377221715004208*

Moscato, V., Picariello, A. & Sperlí, G. (2021), 'A benchmark of machine learning approaches for credit score prediction', *Expert Systems with Applications* **165**, 113986.
URL: *https://www.sciencedirect.com/science/article/pii/S0957417420307636*

Probst, P., Wright, M. N. & Boulesteix, A. (2019), 'Hyperparameters and tuning strategies for random forest', *Wiley interdisciplinary reviews. Data mining and knowledge discovery* **9**(3), n/a.

Purda, L. & Ying, C. (2022), *Consumer Credit Assessments in the Age of Big Data*, Springer International Publishing, Cham, pp. 95–113.
URL: *https://doi.org/10.1007/978-3-031-12240-8*$_6$

Regulation, G. D. P. (2018), 'General data protection regulation (gdpr)', *Intersoft Consulting, Accessed in October* **24**(1).

Solove, D. J. (2006), 'A taxonomy of privacy', *University of Pennsylvania law review* pp. 477–564.

Trivedi, S. K. (2020), 'A study on credit scoring modeling with different feature selection and machine learning approaches', *Technology in Society* **63**, 101413.
URL: *https://www.sciencedirect.com/science/article/pii/S0160791X17302324*

Verma, S. (2019), 'Weapons of math destruction: how big data increases inequality and threatens democracy', *Vikalpa* **44**(2), 97–98.

Wang, G., Hao, J., Ma, J. & Jiang, H. (2011), 'A comparative assessment of ensemble learning for credit scoring', *Expert Systems with Applications* **38**(1), 223–230.
URL: *https://www.sciencedirect.com/science/article/pii/S095741741000552X*

Yang, L. & Shami, A. (2020), ‘On hyperparameter optimization of machine learning algorithms: Theory and practice’, *Neurocomputing* **415**, 295–316.
URL: *https://doi.org/10.1016%2Fj.neucom.2020.07.061*

8 Appendix

8.1 Descriptives

Table 8: Categorical variables description

Table 8 shows the categorical variables that are converted to continuous variables or binary variables. It displays the name of the categorical variable, the different categories, and the value assigned for each category.

Categorical Variables	**Category**	**Continuous**
Municipal group 2023	Rural municipality	0
	Rural municipality with tourism industry	1
	Low-commute municipality close to larger city	2
	Smaller city/town	3
	Commuter municipality near smaller urban areas	4
	Commuter municipality near big city	5
	Commuter municipality near larger city	6
	Metropolitan Cities	7
	Bigger cities	8
	NaN	9
Occupation	Limited	0
	Other	1
	PermanentLessThan6Months	2
	PermanentMoreThan6Months	3
	Retired	4
	SelfEmployed	5
	Student	6
	Unemployed	7
	NaN	8
CivilStatus	Cohabitant	0
	Divorced	1
	MarriedPartner	2
	Separated	3
	Single	4
	Unmarried	5
	WidowWidower	6
	NaN	7
Exposed areas	Not exposed areas	0
	Particularly vulnerable areas	1
	Exposed areas	2
	Risk areas	3
Categorical Variables	**Category**	**Binary**
Role	CoApplicant	0
	MainApplicant	1
granted_loweramount	Granted	0
	GrantedLowerAmount	1
gender	Female	0
	Male	1
New_blancodebt	Have blanco loan at application	0
	Have no blanco loan at application	1
stated_diff_income	Have not Stated higher income than reported in UC_annual income (10000<)	0
	Stated higher income than reported in UC_annual income (10000>)	1
Dependent Variable	**Category**	**Binary**
binary_debtcollection	Is sent to debt collection: FALSE	0
	Is sent to debt collection: TRUE	1

9

8.2 Results from 12m default

Table 9: Class weights performances 12m defaults

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	7335	658	83.99%	90.03%	Balanced	1870	344	82.26%	75.57%
45%	5195	954	87.68%	89.99%	45%	1870	380	85.31%	75.78%
40%	3482	1274	90.47%	90.10%	40%	1025	429	88.35%	75.91%
30%	1381	1894	93.44%	90.63%	30%	498	502	91.99%	76.19%
20%	264	2817	93.83%	81.09%	20%	163	601	93.88%	76.18%
None	0	3027	93.94%	91.14%	None	0	679	94.56%	76.15%

(a) Random Forest

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	15928	852	66.39%	76.09%	Balanced	3809	261	67.39%	70.66%
45%	14672	991	68.63%	75.39%	45%	3809	293	74.52%	71.31%
40%	7064	1627	82.59%	75.44%	40%	1753	394	82.80%	71.07%
30%	3057	2135	89.60%	75.93%	30%	742	515	89.93%	70.84%
20%	1543	2376	92.15%	74.52%	20%	325	593	92.64%	70.24%
None	0	3023	93.94%	75.13%	None	0	680	94.55%	70.60%

(b) Decision Tree

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	14144	883	69.90%	76.94%	Balanced	3809	261	67.39%	70.66%
45%	11134	1120	75.45%	76.92%	45%	3809	265	69.43%	70.97%
40%	4739	2298	85.90%	64.97%	40%	1753	394	82.80%	71.07%
30%	472	2925	93.20%	65.11%	30%	686	520	90.34%	70.83%
20%	2	3026	93.93%	64.41%	20%	319	587	92.74%	70.99%
None	0	3027	93.94%	62.19%	None	0	680	94.55%	70.60%

(c) PLTR

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	14402	901	69.35%	76.13%	Balanced	3607	212	69.40%	74.76%
45%	11273	1143	75.13%	76.13%	45%	3053	247	73.56%	74.78%
40%	4542	2318	86.26%	64.96%	40%	1215	466	86.53%	67.08%
30%	527	2927	93.08%	65.10%	30%	218	600	93.45%	67.01%
20%	5	3026	93.93%	64.36%	20%	6	677	94.53%	66.42%
None	0	3027	93.94%	62.11%	None	0	679	94.56%	63.84%

(d) Penalized Logistic Regression

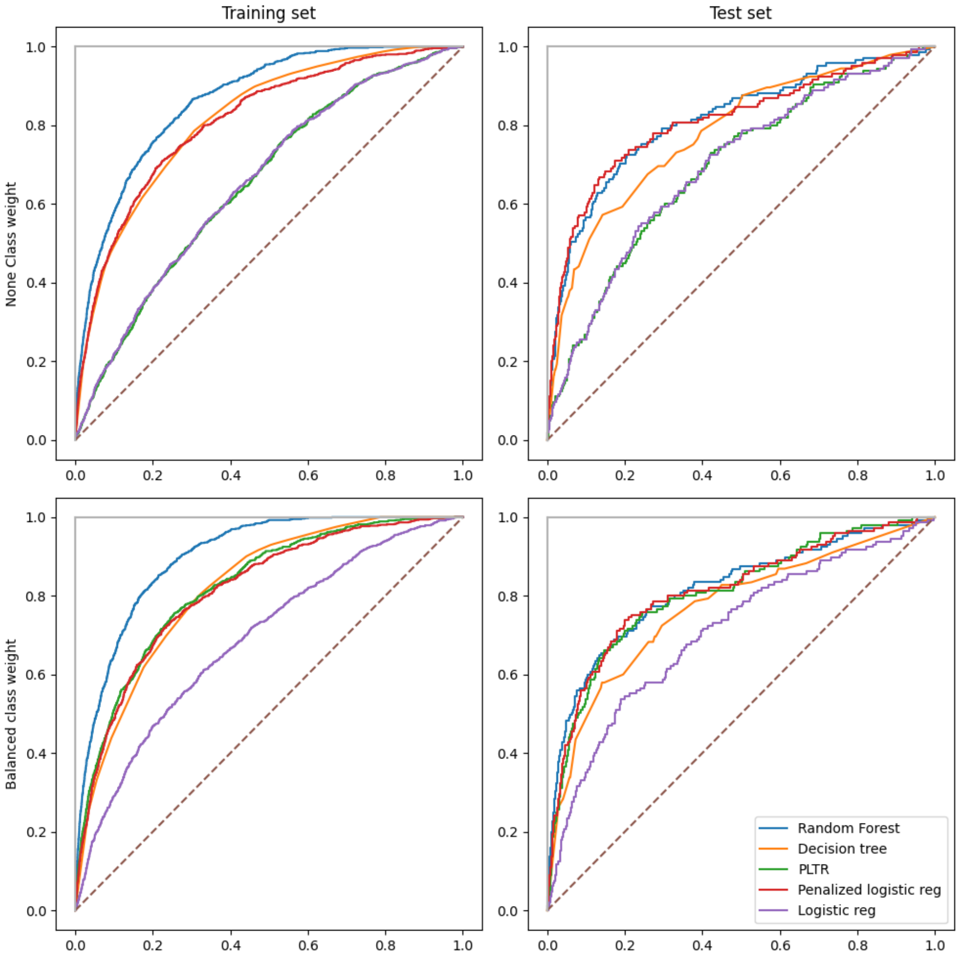
Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	19315	1162	58.98%	63.77%	Balanced	3579	305	68.88%	67.09%
45%	12066	1705	72.41%	64.49%	45%	3579	402	80.59%	67.45%
40%	4543	2317	86.26%	64.96%	40%	752	519	89.82%	67.56%
30%	527	2927	93.08%	65.10%	30%	76	652	94.17%	67.25%
20%	5	3026	93.93%	64.36%	20%	1	680	94.54%	66.39%
None	0	3027	93.94%	62.11%	None	0	680	94.55%	64.07%

(e) Logistic Regression

Table 7 represent the performance measures and the error types for each model and class weight. CW stands for the class weights used. The type I and type II errors represent the number of observations that falls into each error. Where PCC and AUC scores are the performance measures. On the left-hand side, we have the results from predicting on the training set, whereas, on the right-hand side, we have the results from predicting on the test set.

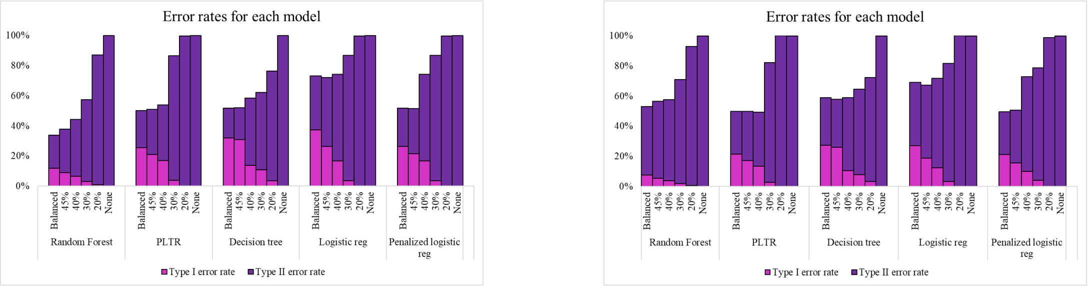
8.3 Results from 5m default

Figure 13: ROC curve results in 5m defaults



The ROC curves in Figure 13 compare the outcomes of using a balanced class weight in the dataset with the outcomes of using no class weights for the 5-month defaults. The top row of ROC curves displays the results obtained by predicting defaults on both the training and test sets without any class weight. In contrast, the bottom row exhibits the results achieved by predicting defaults on both the training and test sets with a balanced class weight.

Figure 14: Resulting Error rates from 5m defaults



(a) training set

(b) test set

The graphs in Figure 14a and 14b show the error rates for the different models and each class weight for the 5-month defaults. The balanced weight is when the different classes are assigned the same importance and the 'none' represent no class weight. The pink area is the type 1 error rate and the purple area represents the type 2 error rate. The sum of these areas represents the total errors for each model and weight.

Table 10: Class weights performances 5m defaults

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	6287	182	88.03%	92.73%	Balanced	1009	66	92.05%	81.80%
45%	4715	238	90.84%	92.67%	45%	1009	74	94.10%	81.85%
40%	3451	311	93.04%	92.71%	40%	516	78	95.61%	82.04%
30%	1637	448	96.14%	92.92%	30%	266	100	97.29%	82.13%
20%	436	711	97.88%	83.78%	20%	71	134	98.48%	81.06%
None	0	823	98.48%	87.61%	None	0	145	98.93%	81.50%

(a) Random Forest

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	17074	163	68.12%	81.89%	Balanced	3653	46	72.63%	76.85%
45%	16448	173	69.26%	81.89%	45%	3653	46	73.82%	76.85%
40%	7314	369	85.79%	81.88%	40%	1411	70	89.04%	77.42%
30%	5793	423	88.50%	81.92%	30%	1058	82	91.57%	78.08%
20%	1857	599	95.46%	82.11%	20%	433	100	96.06%	77.54%
None	0	823	98.48%	81.79%	None	0	145	98.93%	77.23%

(b) Decision Tree

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	13599	202	74.47%	82.30%	Balanced	2872	41	78.45%	80.69%
45%	11203	246	78.82%	82.31%	45%	2872	47	82.49%	80.74%
40%	8991	304	82.81%	82.30%	40%	1797	52	86.32%	80.82%
30%	2128	679	94.81%	70.27%	30%	375	115	96.37%	71.92%
20%	57	820	98.38%	69.96%	20%	6	145	98.88%	72.04%
None	0	823	98.48%	65.42%	None	0	145	98.93%	69.43%

(c) PLTR

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	14066	209	73.59%	81.37%	Balanced	2623	41	80.29%	81.07%
45%	11438	247	78.39%	81.37%	45%	2623	50	84.29%	81.11%
40%	8894	474	82.67%	69.91%	40%	1665	86	87.04%	71.51%
30%	1971	685	95.09%	70.26%	30%	425	114	96.01%	71.89%
20%	50	819	98.39%	69.96%	20%	4	145	98.90%	72.03%
None	0	823	98.48%	81.18%	None	0	145	98.93%	81.11%

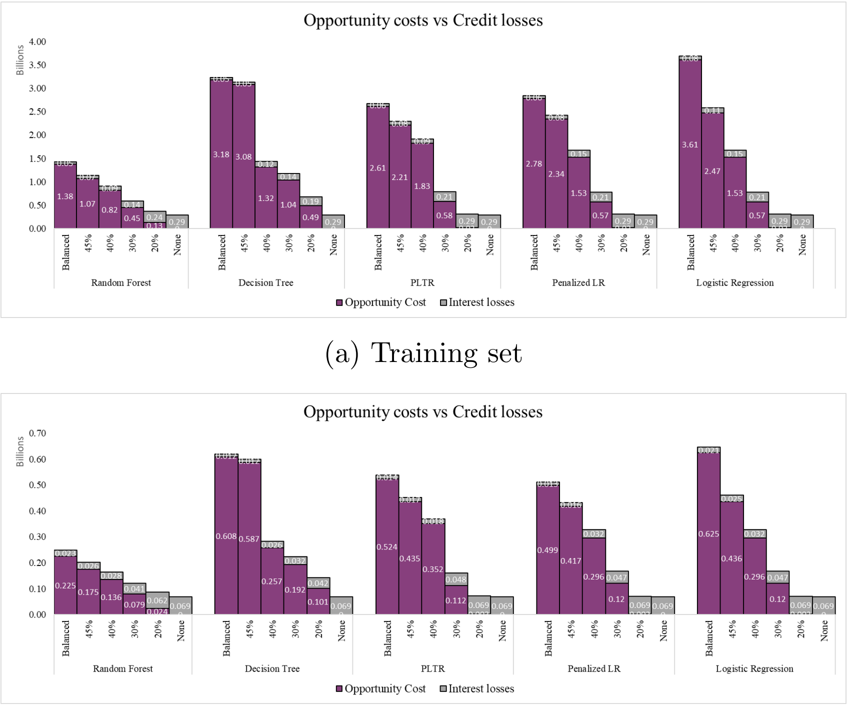
(d) Penalized Logistic Regression

Training set Results					Test set Results				
CW	Type I error	Type II error	PCC	ROC AUC	CW	Type I error	Type II error	PCC	ROC AUC
Balanced	19844	296	62.75%	68.96%	Balanced	3630	61	72.69%	70.82%
45%	14006	376	73.40%	69.51%	45%	3630	70	80.84%	71.22%
40%	8892	474	82.68%	69.91%	40%	1665	86	87.04%	71.51%
30%	1971	685	95.09%	70.26%	30%	425	114	96.01%	71.89%
20%	50	819	98.39%	69.96%	20%	4	145	98.90%	72.03%
None	0	823	98.48%	65.61%	None	0	145	98.93%	69.49%

(e) Logistic Regression

Table 10 represent the performance measures and the error types for each model and class weight, predicting 5-month defaults. CW stands for the different class weights. The type I and type II errors represent the number of observations that falls into each error. Where PCC and AUC scores are the performance measures. On the left-hand side, we have the results from predicting on the training set, whereas, on the right-hand side, we have the results from predicting on the test set.

Figure 15: Opportunity Costs vs Credit Losses 5m defaults



(a) Training set

(b) Test set

Figure 15 displays the resulting credit losses and opportunity costs for the defaults within 5 months. Again, the opportunity cost is represented by the purple field, and the losses are represented by the grey field. In both of the samples, each model has resulting opportunity cost and losses for each class weight and model. The y-axis represent the amount of total losses, expressed in billion SEK.