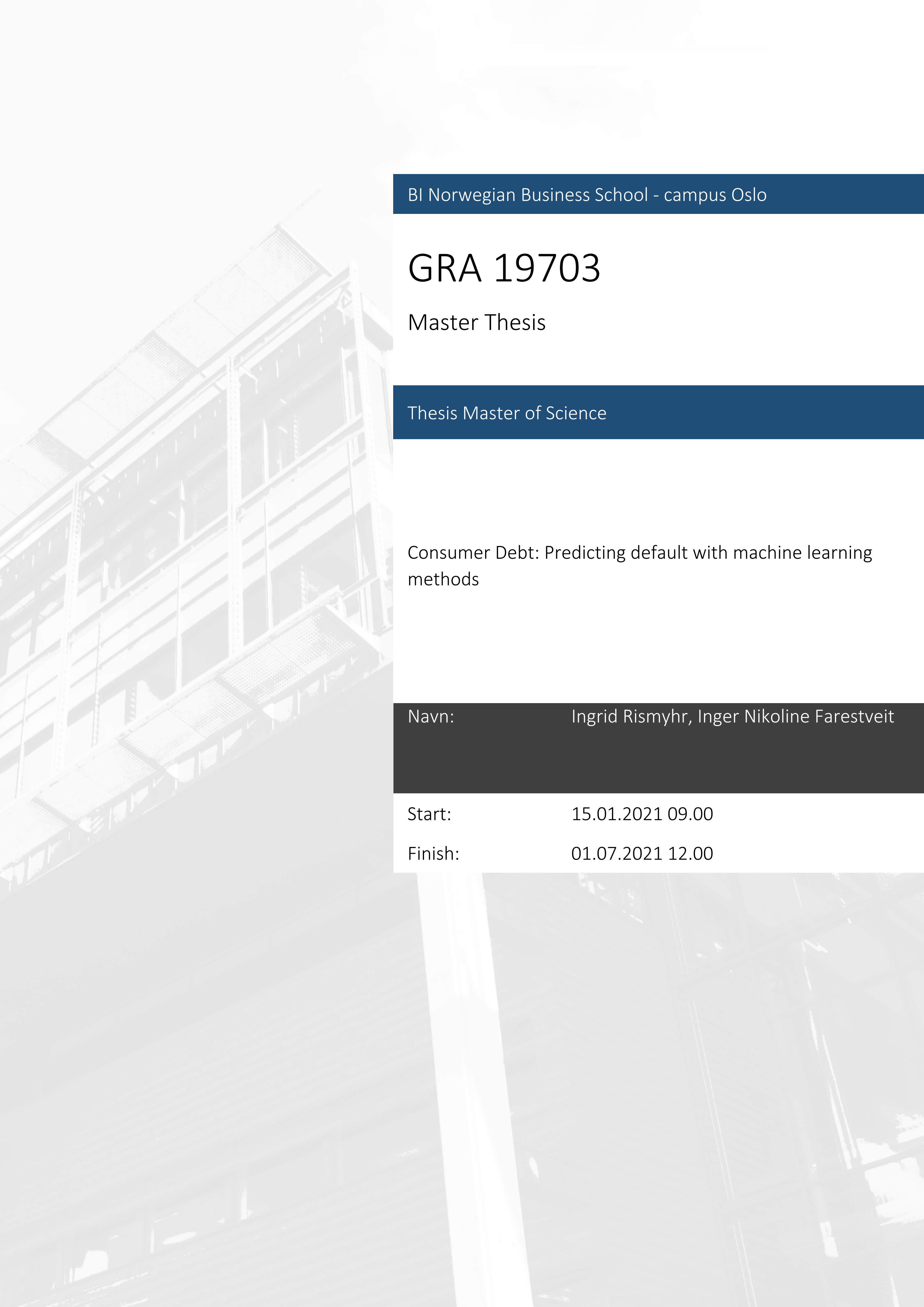


BI Norwegian Business School - campus Oslo

GRA 19703

Master Thesis

Thesis Master of Science

Consumer Debt: Predicting default with machine learning methods

Navn: Ingrid Rismyhr, Inger Nikoline Farestveit

Start: 15.01.2021 09.00

Finish: 01.07.2021 12.00

BI NORWEGIAN BUSINESS SCHOOL
Oslo, Spring 2021

CONSUMER DEBT

Predicting default with machine learning methods

Inger N. Farestveit & Ingrid Rismyhr

Supervisor: Genaro Sucarrat

Master Thesis GRA1970

Business Analytics

BI NORWEGIAN BUSINESS SCHOOL

Campus Oslo

Acknowledgements

This thesis is written as part of the Master of Science program in Business Analytics at BI Norwegian Business School. Any errors are the sole responsibility of the authors.

We would like to thank Kantega, Sparebank-1 Kreditt and Norsk Gjeldsinformasjon for valuable data, good cooperation, and guidance. Furthermore, we would like to send our sincere gratitude to Jens Morten Nilsen and Christer S. Dale from Sparebank-1 Kreditt for exceptional help and guidance beyond what is expected. Moreover, we would like to thank our supervisor Genaro Sucarrat for good discussions, valuable input, and guidance throughout the process. Finally, we would like to thank our family and friends for their patience and support. A special thank you also goes out to everyone who has proofread the thesis and inspired us along the way.

Disclaimer. One of the authors, Inger N. Farestveit, was employed at Kantega when writing the thesis.

BI NORWEGIAN BUSINESS SCHOOL

Oslo, June 2021

Inger N. Farestveit

Ingrid Rismyhr

Abstract

The aim of this thesis is to explore if a machine learning model can create value by predicting default at the time of credit application. In extension of this, the thesis will evaluate whether a predictive model can be used to reduce future monetary losses associated with accepting applicants who later default on their consumer debt. Furthermore, we explore whether or not information from the Norwegian Registry of Consumer Debt improves the predictive performance.

The scope of the thesis is limited to customers in the Norwegian market **who was granted** consumer debt by the examined company in the period of November 2019 - February 2020. Several resampling techniques as well as cost-sensitive learning were explored as the data was highly imbalanced. The issue was ultimately addressed with cost-sensitive learning, by assigning weights to the classes. The following machine learning (ML) models were explored: ML version of Logistic Regression, Random Forest and eXtremeGradientBoosting. These models were optimized and compared with traditional statistical models. The models were trained on a stratified random selection consisting of 85% of the data. The results were obtained by deploying the model on the remaining 15% of the data, called the holdout data. The ML models were individually optimized across three dimensions: variable selection, hyperparameter tuning, and resampling technique. Ultimately, the best performing model was eXtremeGradientBoosting trained on data with no resampling, 66 variables and a minority class weight of 36:1.

The study concludes that a machine learning model can create value by predicting default at the time of credit application, as 44% of the applicants who defaulted were predicted correctly. This comes at the expense of a 4% misclassification of applicants who did not default. However, monetary losses are reduced as the avoided loss exceeds the potential loss of income. Additionally, the information from the Norwegian Debt Registry contributed to an increase in performance by correctly predicting more defaults.

Keywords – Machine Learning, Consumer Debt, Debt Registry, BI

Contents

Figures

Tabels

1 Introduction

This chapter provides information regarding the background of the thesis as well as the potential benefit, the research question, and the scope. The final subsection of the chapter provides an explanation of all defined terms used in the thesis (table 1.1, 1.2).

1.1 Background

The level of consumer debt[1] amongst the Norwegian public has been high over the past years. One in three customers with repayment loans experienced difficulties with managing their loan repayments (Haugan, 2020). Due to the lack of exchanged information amongst the debt providers, several claim that it is too easy to obtain consumer debt (Barne- og familiedepartementet, 2019). Prior to July 2019, the financial institutions did not have sufficient information about the applicants, which made it possible to obtain high levels of debt from multiple providers. Along with the continuous growth in consumer debt, this led to new legislation concerning lending practices. Hence, financial institutions operating in the Norwegian market are now obligated to report all consumer debt to a registry. The purpose of the law is in short to facilitate a safe and effective registration and exchange of information, to prevent debt problems among private citizens (Gjeldsinformasjonsloven, 2017, §1). The registry has seemingly contributed to lower amounts of consumer debt in total. However, there unfortunately has been an increase in the share of citizens who default on their consumer debt (Finans Norge, 2021a).

1.2 Research Question

The goal of the master thesis is to explore whether Sparebank-1 Kreditt (from here on referred to as the Company) could diminish the potential loss related to defaulting applicants by using a predictive model. The losses in this instance are related to accepting a customer who will default on their debt within the

[1] In this thesis the term consumer debt refers to all unsecured debt in Norway, which includes repayment loans, credit facilities and charge cards.

next 12 months. Hence, the research question is:

Can a machine learning model create value by predicting default at the time of credit application?

1.2.1 Subquestions

To further create value, the following two subquestions were explored:

1. *Can the predictions be used to reduce future monetary loss?*
2. *Does data from the Debt Registry increase the predictive performance?*

1.3 Scope

The scope of this thesis is to evaluate the Company's credit customers in the Norwegian market. The prediction model has been developed with data from $12,817$ applicants who were **granted** unsecured debt by the Company in the period November 2019 - February 2020. This data is of the type panel-data, as it consists of static information about each applicant at the time of the application, and a binary indicator of whether or not the applicant default within 12 months. Panel-data is a combination of cross-sectional data and time-series data, which allows one to observe one applicant over a period of time (Sucarrat, 2017, p. 43).

1.4 Definitions

The most commonly used terms in the thesis are defined below.

Term	**Definition**
Default	Monetary claim sent for debt collection 90 days after first invoice, due to failed payment.
Consumer debt	Unsecured debt including repayment loans, credit facility and charge cards.
Total debt	All debt including mortgages, vehicle loans, Consumer Debt and student loans.
Credit Facility	Credit card with a given credit limit where payments can be limited to a minimum amount.
Repayment loan	Short term loan of unsecured debt.
Charge Card	Credit card without explicit credit limit where all debt is to be payed in full once a month.
Defaulting Applicants	Applicants who will default within 12 months.
Non-Defaulting Applicants	Applicants who will not default within 12 months.
Debt Registry	The Norwegian Registry of Consumer Debt provided by Norsk Gjeldsinformasjon.
The Company	The examined company, Sparebank-1 Kreditt

Table 1.1: Definitions part 1

Term	**Definition**
The Credit Agency	An external agency who provides credit evaluations.
The Market	The study case market which is limited to Norwegian consumer debt.
Loss given Default	The loss associated with a customer defaulting on their Consumer Debt
Potential loss of income	The potential loss associated with declining a Non-Defaulting Applicant
Missed Income	The loss associated with a Non-Defaulting Applicant who has been declined.
The Market	The study case market which is limited to Norwegian consumer debt
Loan Product	A credit agreement which referrers to either a repayment loans, credit facility or charge cards.

Table 1.2: Definitions part 2

2 Context

This chapter provides detailed information about the current situation and regulations in the market. Additionally, the chapter provides insight into the current application process within the Company.

2.1 The Market

Norwegian household debt has rapidly increased over the years and was in 2019 claimed by the Norwegian government to be historically high (Regjeringen, 2019). Despite this reality, consumer loans only consist of approximately 3% (152,6 billion NOK [2]) of the total debt as mortgages account for the majority. Nevertheless, there has been an extensive focus on regulating the lending practices for consumer loans, as these loans have a vastly higher interest rate, and the growth over the several years has been twice as high as the general growth in debt. Due to the high interest rates, consumer loans accounts for approximately 14% of a household's total interest expenses (Regjeringen, 2019). Additionally, one in three customers who have repayment loans experience difficulty paying their repayment amount. These customers account for over 60% of the total amount of repayment loans in Norway (Haugan, 2020). It is important for the society that the household debt is sustainable, which has increased the need for regulations and information-sharing amongst the financial institutions. Whereas the lending practices for mortgages have been regulated since 2015, the first regulations of consumer loans emerged in 2017 (Finanstilsynet, 2019), (Regjeringen, 2019).

2.1.1 Regulations

As a measure to reduce the high increase of consumer loans, the first guidelines for lending practices along with regulations concerning the marketing of consumer loans were established in 2017.

The marketing regulations emphasized that debt providers are prohibited from advertising the accessibility of their loans. This included, but was not

[2] As of May 31st (Norsk Gjeldsinformajson, 2021)

limited to, the simplicity of the application process or how quickly credit could be granted. Nevertheless, the debt providers could provide relevant information regarding the processing time, the application process, and their conditions, but this information should not be more prominent than other important information like the cost of credit (Regjeringen, 2017).

However, as the guidelines from 2017 concerning lending practices for consumer loans were not formally regulated, some financial institutions chose to not completely follow them. Consequently, the Ministry of Finance induced a regulation in February 2019 that concerned legal requirements for lending practices on consumer loans. The regulation considered the customer's capacity to manage their debt and stated that their total debt to annual income ratio could not exceed 5:1. Additionally, the regulations set specific requirements for instalment payments and allowed a flexibility quota of 5%. In order for the financial institutions to fulfill the requirement, they potentially had to check each consumer in a debt registry, which was introduced the same year (Regjeringen, 2019), (Finanstilsynet, 2020, p. 8).

2.1.2 Debt Registry

There were various opinions regarding whether a debt registry should be implemented or not. Whereas many actors wanted a registry, Datatilsynet on the other hand were against the proposed implementation. One of their main arguments was that it would be intrusive and an invasion of the consumer's privacy. Especially when the majority of the consumers could service their Consumer Debt without problems. Instead, they suggested other preventive measures like an interest rate ceiling (Thon, 2014).

Nevertheless, the debt registry was established in July 2019. This registry enabled the financial institutions to make more thorough credit assessments, which could further prevent insolvency. From July 2019 until the end of September 2020, there were in total 3 companies (Gjeldsregisteret AS, Norsk Gjeldsinformasjon AS & Experian Gjeldsregister AS) with a license to serve as a debt information company. The debt registry includes various types of unsecured debt, which can be divided into 3 subcategories: Credit Facility, Repayment Loans and Charge Cards (Finanstilsynet, 2020, p. 14).

2.2 The Company

The Company is part of an alliance which consists of in total 15 independent banks across the country, who collaborate on some aspects of their operations (Sparebank-1, n.d.). The Company manages the majority of Consumer Debt for all banks within the alliance with a few exceptions. In time, it is planned for the Company to manage the alliance's entire portfolio of Consumer Debt.

2.2.1 Application Process

The information presented is based on information received during an interview with a representative from the Company on 29th of January 2021.

Whether an applicant applies for a new Consumer Loan or an increased credit limit, the application process starts with an application form. The form is either filled out online, in the mobile application or in person with a bank advisor. The questions in the form are developed to capture an applicant's current- and potential economic situation. Hence, the form consists of several questions regarding income, debt, expenses, and life situation such as employment, age, and marital status. The majority of the information provided by the applicant will be based on trust. However, some of this information is cross-checked with other external data sources. For instance, income and tax are validated based on external information from a Credit Agency. Additionally, they acquire information about the applicant from public registries such as the Debt Registry, the population registry, the motor vehicle registry, and the real estate registry. If the applicant is a prior customer of the alliance, the Company could, with consent, also cross-check customer information and history from internal registries.

To assess the potential risk associated with an applicant, the Company also acquires a score from a Credit Agency which provides probability estimates of how likely it is that the applicant will receive a payment remark in the next 24 months. Additionally, the Company has started the process of developing their own scoring model, in the form of a machine learning algorithm, which sends out a warning if an applicant applies for more credit than they can sufficiently operate (Kantega, n.d.). Furthermore, to uphold both internal policies and governmental regulations, the Company also makes calculations regarding liquidity and debt ratio expenses. Today, the most common reason for declining an applicant is unsatisfactory liquidity regarding national regulatory requirements. Additionally, applicants are instantly declined if they have incurred a payment remark.

The distinction between debt collection and payment remarks must be noted. The Company does not have access to any ongoing debt collection cases as this is not public information, compared to payment remarks. Therefore, it is possible for one applicant to have a number of debt collection cases without the bank knowing at the time of application. Due to regulatory requirements, it often takes a long time before a debt collection case actually leads to a payment remark.

2.2.2 Default Process

The information presented is based on information received during an interview with a representative from the Company on 29th of January 2021. The number of days displayed in figure 2.1 is based on approximation.

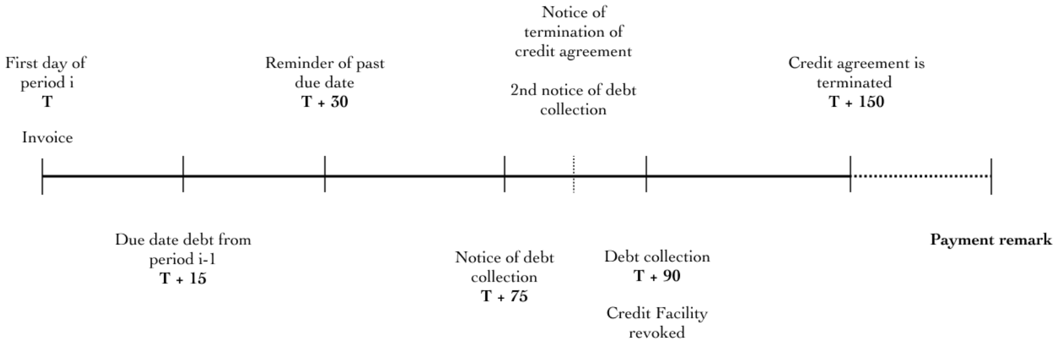

Notice of termination of credit agreement
First day of period i T
Reminder of past due date T + 30
2nd notice of debt collection
Credit agreement is terminated T + 150
Invoice
Due date debt from period i-1 T + 15
Notice of debt collection T + 75
Debt collection T + 90
Credit Facility revoked
Payment remark

Figure 2.1: Process of debt collection and payment remark.

The process from invoice to default to payment remark starts when the customer fails to pay their invoice at (T+15) as illustrated in figure 2.1. The customer at this point, had a maximum of 45 days referral of payment. For instance, if the bill for the prior month is invoiced the 1st of the month, it will be due the 15th of the month. If the customer does not fully pay the outstanding balance, they could choose to pay a minimum amount, which consists of a small percentage of the total outstanding balance. However, if the customer does not pay, then they will receive a reminder the following month (T+30). If they still choose not to pay after the first reminder, they will receive a notice of debt collection two months after the original due date (T+75). Then almost in parallel, they will receive a notice of termination of the credit agreement and a second debt collection notice. If the customer does not pay after the second notice of debt collection, the debt goes to debt collection two months after the first notice of debt collection (T+90).

After the debt is sent for debt collection, the Credit Facility is revoked. The customer still has two months to pay an amount $\geq$ the minimum amount. If this occurs at any time during the process, the customer is declared as a regular customer, and their card is reopened. However, if the customer does not pay the minimum amount approximately within 60 days after the debt is first sent for collection (T+150), then the Company's policy is to terminate the credit agreement.

In conclusion, the customer has several chances, over an extended period of time, to pay their debt before they end up with a payment remark[3]. Payment remarks are more severe as they are the end result of failed debt collection. A payment remark has severe consequences, as it will impact the customer's credit score and make it difficult to get future credit applications approved. As mentioned, the Company does not grant credit to applicant's who currently have payment remarks.

[3] A payment remark is deleted either when the full amount is paid (including interest), or after 4 years unless the debt collection agency initiates new legal steps in order to register a new payment remark. (Lindorff, n.d.).

3 Data

This chapter provides information regarding the three data sources used in the thesis. The main source of data that was used to develop a predictive model consists of **granted applications** from the Company **in the period November 2019 to February 2020,** and the respective target (Default/Non-Default) within 12 months. It must, however, be stated that the world was undergoing a pandemic and a global crisis during the year in question, which may have impacted the outcome of whether the applicants Defaulted or not. Therefore, aggregated data about the number of granted credit applications in the Company were examined for an extended period of time (January 2019 - April 2021). Additionally, data from the Debt Registry has been used to state the Company's market share in the Norwegian Market. It should be noted that, from the data sources presented below, source 1 is the only data used for the predictive model. Sources 2 and 3 are only included for illustrative purposes.

3.1 Source 1: Application Data

The main source of data is from the period of November 2019 to February 2020, and contains $12,817$ observations, where one observation is an applicant who was **granted** Consumer Debt by the Company. One observation also consists of static information about the applicant at the time of the application as well as a binary indicator of whether or not the applicant defaults within 12 months. This is the data used for training and developing the prediction model in the thesis.

3.1.1 Variables

The variables in the application data include all information available for the Company at the time of application. Some of the variables are collected from the following external registries: the motor vehicle registry, the tax authorities, the population registry, and the Debt Registry. Additionally, if the applicant is a former or an existing customer, the Company could have access to some internal data provided by banks within the alliance. The applicant also selfregisters some of the variables where some of them are checked against external registries while others are based on trust. Furthermore, the data includes a binary indicator that states whether or not a given applicant has defaulted within the first year. Additionally, the data includes, but is not limited to, wealth, income, preexisting unsecured debt and mortgages, employment, marital status, registered vehicles, type of employment, number of children. The data is anonymous and the only personal characteristic of an applicant is their year of birth. All variables can be seen in appendix A2.1 and A2.2. There are in total 64 original variables and 12,817 rows, where one row represents one application.

3.2 Source 2: Aggregated Data

The second source of data contains the total number of applications received in the period of January 2019 to April 2021. Hence, this data contains aggregated information about applications 10 months prior to, and 14 months subsequent to the main source of data. The aggregated data consists of 28 observations, where one observation is one month and contains the total number of granted and declined applications for the specific month. As the **main source of data only consisted of granted applications**, the average rejection rate in both periods was of interest. To evaluate if the number of granted applications in the main source of data was representative for the extended period, the average rate of declined applications in the limited period was compared to the average rate in the extended period.

In the extended period, the Company received on average ≈ 7,000 applications each month. The period for the main source of data is marked in red (figure 3.1), and the average number of applications received in this period was ≈ 7,200. As can be seen from the figure, there are continuous fluctuations in the number of applicants, but the largest drop occurs after the national lock down in March - April 2020, which naturally lowers the overall average when assessing the extended period.

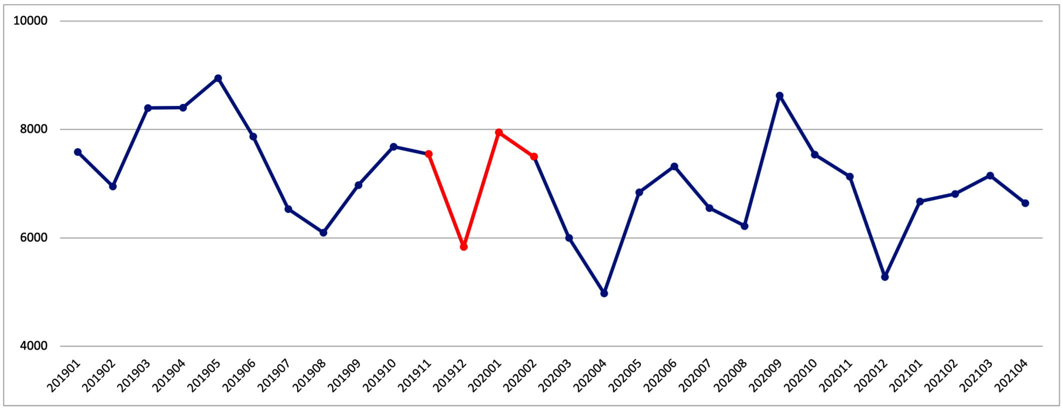


Figure 3.1: Total number of applicants

In the extended period ≈ 3.4% of the granted applicants Defaulted[4], whereas in the main source of data, only 2.7% Defaulted. The rate of rejected applications can be seen in figure 3.2, where the period from the main source of data is marked in red. The average rejection rate of all applications in the extended period is 43% with a standard deviation (σ) of 3.6%. The average rejection rate for the main source of data is 45% which, according to the Company, is representative for the extended period. This claim was substantiated as the average of the period from the main source of data is only $0.6 \times \sigma$ higher than the average of the extended period.

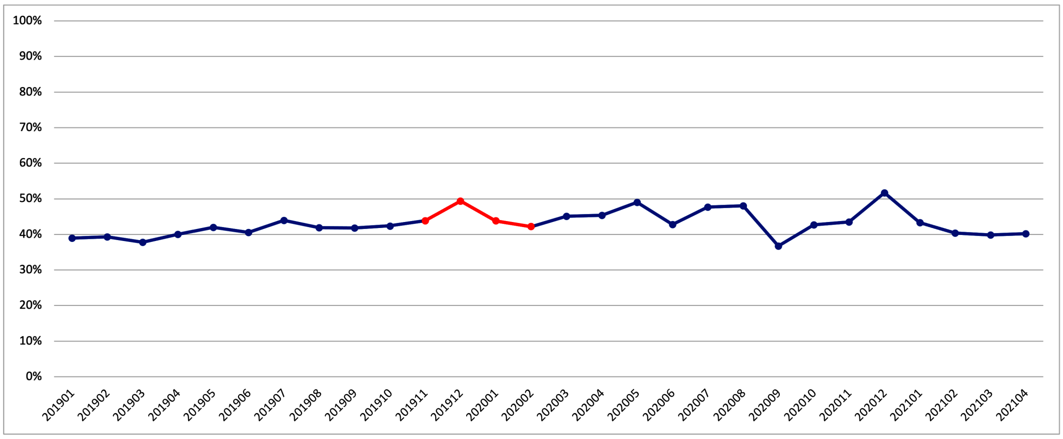


Figure 3.2: Rejection rate of all applications

[4] Estimate provided by the Company.

3.3 Source 3: The Debt Registry

The third source of data presented in this section provides additional information about the Company's total portfolio of Consumer Debt[5]. This information is presented to provide an overview of the Company's market share and the amount of loan products associated with their customers.

The Company manages $\approx 4\%$[6] of the total consumer debt in the Norwegian market (154,8 billion NOK[7]). As can be seen from figure 3.3, $\approx \frac{1}{3}$ of the customers only have 1 unsecured loan product, which means that The Company is their sole provider of unsecured debt.

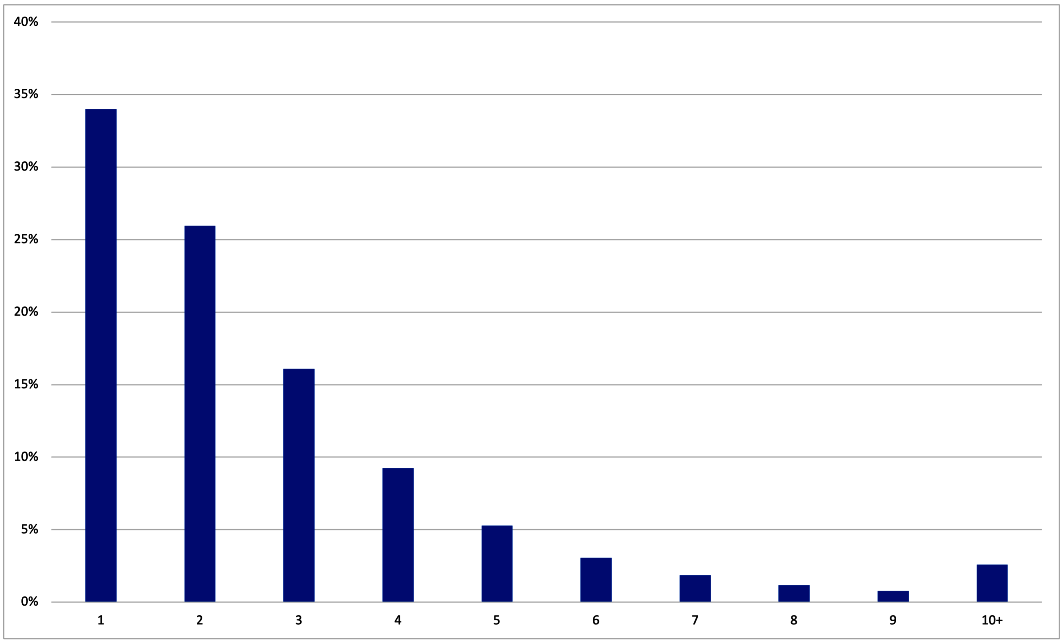


Figure 3.3: Total amount of loan products

Furthermore, the majority of the Company's customers have 1-3 loan products in total, where the loan products granted by the Company ≥ 1. However, when assessing the entire customer portfolio, their customers have on average ≈ 3 loan products.

[5] The information is retrieved directly from the Debt Registry.

[6] As of May 14th 2021.

[7] This information is retrieved directly from a database at the Debt Registry on May 14th 2021. The amount differs slightly from the amount in section 2.1 Market, as this is based on public information published on May 31st by the Debt Registry (Norsk Gjeldsinformajson, 2021).

3.4 Data Preparation

The data was explored, prepared, and encoded to remove potential errors and misleading information. In this thesis, the data preparation has been divided into the following steps; exploratory analysis, preprocessing, and variable transformation. Prior to the data preparation, the data consisted of 66 columns and 12,817 rows. After the preparation was conducted, the data contained 112 columns and 12,794 rows.

3.4.1 Exploratory Analysis

Errors were discovered in the first two versions of the data, which were corrected by the Company in the finalized data set. Additionally, observations with duplicated, missing, or inaccurate values were removed. This resulted in the removal of 6 observations on account of age = 0, as well as 17 duplicated rows. Finally, descriptive statistics was displayed for all variables, which revealed that several consisted solely of the value 0. All variables containing only the value 0 were removed.

All variables were systematically examined and evaluated. However, we are humble to the fact that there might be undetected errors. The modifications of the data are further explained in the following sections.

3.4.2 Preprocessing

The knowledge obtained from the exploratory analysis regarding minor sources of data pollution were the basis for how the data was further prepared. Debt within the Company displayed as a negative number was set to the absolute value. According to the Company, the value -1 indicated "no information was found" in external registries. However, it was evident that -1 had individual meaning for each variable. Hence, four binary variables were created (table 3.1) to correctly capture the underlying information of -1 in each variable, while the respective values in the original variables were changed to zero.

Binary variable	**Explanation**
External Applicant	The applicant has no prior customer relationship with the Company or any of its partners
Not in Debt Registry	There is no unsecured debt connected to the applicant in the Debt Registry[8]
Missing Tax Information	No tax information available about the applicant[9]
Missing Days Since Move	Applicants with no prior registered address[10]

Table 3.1: Binary variables created from categorical variables.

3.4.3 Variable Transformation

Implicit information in the data was used to transform variables and create new variables, these are illustrated in table 3.2.

Binary variable	**Explanation**
Application Weekend	Application filed Saturday or Sunday
Application Night	Application filed between 23:00 – 6:00
Active Card Deviation	Discrepancy between self reported amount of active credit facilities and registered credit facilities in the Debt Registry
Vehicle Loan No Vehicle	Applicant with vehicle loan $>$ 0 and no registered ownership of vehicle

Table 3.2: Binary variables created from implicit information in the data.

During the exploratory analysis, it was evident that some applicants

[8] Applicable for 39% of the applicants. The technical team responsible for the Debt Registry assumed that these applicants had no prior debt in the registry. The assumption was confirmed by the Company.

[9] Applicable for 5% of the applicants who may be exempt from taxation due to low income or be subject to unknown error at the tax authorities.

[10] The median age of these applicants were 18, which may indicate that they were still registered at their parents address at the time of application.

were registered to have vehicle loans without owning a vehicle. *Vehicle Loan No Vehicle* could indicate that the applicant previously was forced to sell a mortgaged vehicle for a price < remaining loan amount.

Furthermore, the application form required the applicants to state their total number of active credit facilities. *Active Card Deviation* capture any discrepancy between the self-registered amount and the total number of credit facilities registered to the applicant in the Debt Registry.

To distinguish between applications filed during the day and night, the hours were categorized with the creation of a binary variable, which assumed the value 1 if the application was filed between the hours of 23:00 and 6:00. The same logic was applied to capture whether an application was filed on a working day or during the weekend with the binary variable *application weekend*.

The scales of categorical variables were assessed to determine whether to use dummy encoding or label encoding. The ordinal[11] variables were considered to be label encoded while dummy-encoding was used for the nominal[12] variables. However, it was ultimately decided to pursue dummy encoding for all categorical variables. For a variable that could assume m values, m binary variables were created. The models sensitive to multicollinearity[13] (dummy-trap) were trained on *m-1* of the binary variables. This applied to the Linear Probability Model and Logistic Regression, whereas decision trees are robust to handle multicollinearity as only one perfectly correlated variable is chosen when the tree is split (Badr, 2019b).

The following variables were dummy encoded: product name, type of employment, habitation type, marital status, income category, wealth category, and consumer loan category. The data preparation resulted in a dataset which contained 112 columns and 12,794 rows.

[11] The input has a natural ranking

[12] Variable input with no natural ordering

[13] When the last binary variable can be predicted perfectly as it is an exact linear combination of the others (Sucarrat, 2017).

4 Empirical Analysis

Traditional statistical models were compared with machine learning models which were deployed with both standard- and optimized parameters. The data was first partitioned into two subsets (training & holdout) where all models were trained on the training data and the results of the models were obtained on the holdout data. Due to the imbalance amongst the classes in the data, the sample mean of y ($\overline{y}$, where $y =$ Default), was used as the threshold for the traditional models. For the machine learning models, the classes in the training data were assigned weights to compensate for the sample imbalance. However, as there exist a variety of different resampling methods, several techniques were explored before weights were chosen (Brownlee, 2020a, p. 104). The selection process is described in the following section. Ultimately, XGBoost was the best performing model to predict Default, and the results obtained with this model are further discussed in Chapter 5.

4.1 Data Partitioning

Two common methods utilized to minimize the risk of over-fitting[14] a model is k-fold cross validation and holdout. However, due to the imbalance in the data, the holdout strategy was considered more favorable as the model has more instances of the minority class to train, learn, and fit on.

The holdout validation splits the data into minimum 2 folds; train and holdout. The goal of the method is to train the model on one part of the data and then validate the results using the holdout. The method could be extended into 3 folds; train data, validate data, and holdout data. See figure 4.1. The benefit of utilizing 3 folds is that the model can be tuned based on the results from the validation data without the risk of overfitting (as these results can then be validated on the holdout data). With 2 folds, alterations cannot be made to the model based on the results obtained on the holdout data, as there is no data left to validate the changes on (Bronshtein, 2017), (Brownlee, 2020d).

[14] An over-fitted model is tuned to fit the training data too such an extent that it performs poorly on new data (Al-Masri, 2019).

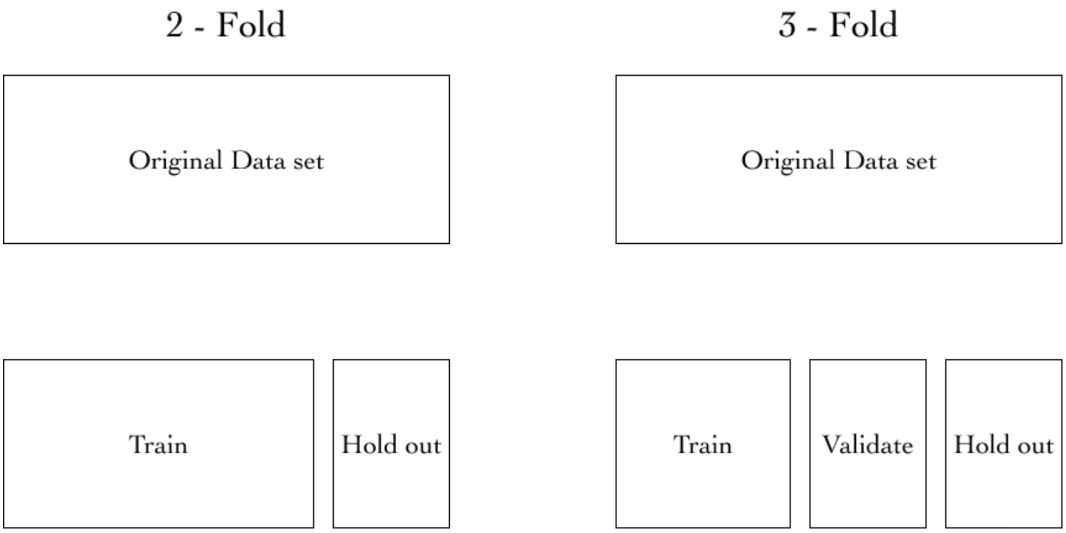


Figure 4.1: Data partitioning using the holdout strategy

Nevertheless, due to the imbalance in the data (2.7% Default) a 2-fold split was considered more beneficial as it allowed more of the minority class to be part of the holdout data without compromising the training data. Although a 3-fold split would be optimal to ensure a final valuation set, a 2-fold split was considered prominent due to the low level of minorities. Thus, a 2-fold stratified[15] split, with 85% of the data in the training and 15% in the holdout, was chosen to ensure a minimum level of minorities in each fold. The data was split with the use of the train_test_split function from the scikit-learn library (scikit-learn 0.24.2, 2021). This function was executed using a stratified split, which splits the data randomly while ensuring that both the training- and holdout data preserve the underlying distribution of the target variable (Default). This resulted in a training data with 10,874 observations (360 Defaults) and a holdout data with 1,920 (54 Defaults). Consequently, the chosen split was a trade-off between giving the model enough data to both train and test on to achieve optimal results and having the ability to validate them. The validity of the results is a prerequisite to ensure that the deployment decision is sufficiently supported. To ensure validity with a 2-fold split, the models were not tuned after deployed on the test data.

[15] To ensure that the unbalanced data preserved the same class distribution(Brownlee, 2020d).

4.2 Handling Imbalanced Classes

Imbalance in the training data [16] could lower the predictive performance of the machine learning models, as these assume a balanced distribution of the classes. In some instances, collecting more data could help diminish the imbalance. However, for this data, the imbalanced is the property of the domain, and not caused by biased sampling or measurement errors. Thus, collecting more data would merely generate larger but equally imbalanced data (Brownlee, 2020a, p. 105). There are various methods for handling imbalance, where no technique is considered universally superior (Brownlee, 2020a, p. 104). Therefore, it was deemed prominent to disclose all considered methods, as they could impact the predictive performance. Ultimately, assigning weights through cost-sensitive learning (section 4.2.2) to the machine learning models was the most efficient way of handling the imbalance. The training data was therefore not subject to any resampling techniques, which is discussed in the following section as it may be considered unconventional.

4.2.1 Resampling

The following resampling techniques were initially considered for handling the imbalance. However, as previously mentioned, none of the resampling techniques further described in this subsection were utilized.

4.2.1.1 Oversampling

The simplest form of oversampling is the Random Over-Sampler (ROS). This technique does not utilize heuristics, but creates balance in the training data by randomly duplicating the minority class numerous times. However, solely duplicating existing instances would not add new information to the model and the technique was therefore not considered. Thus, techniques that utilized heuristics were evaluated. The Synthetic Minority Oversampling TEchnique, known as SMOTE, generates new samples that fit a line between two instances which are close in variable space. There exists two modifications of SMOTE which were both considered (Brownlee, 2020a, p. 122).

[16] Holdout data should not be subject to modifications.

The first modification, Borderline-SMOTE, only oversamples the instances of the minority class which are misclassified and thus more important as they lie on the borderline between the two classes. The second modification, Adaptive Synthetic Sampling, known as ADASYN, generates synthetic instances based on the density of instances in the minority class. If the density is low, the samples are harder to learn, which results in more synthesized instances (Brownlee, 2020a, p. 130, 134). Ultimately, SMOTE was chosen as the oversampling technique.

Even though oversampling might establish a more balanced training set, the disadvantage of using oversampling techniques is that these techniques do not consider the majority class. Furthermore, as the data is severely skewed, the high level of replication could cause the algorithm to over-fit (Brownlee, 2020a, p. 114).

4.2.1.2 Undersampling

The simplest form of under-sampling is the Random Under-Sampling (RUS) technique, which creates a balance in the training data by randomly removing data points from the majority class. However, with severely imbalanced data, this technique could potentially remove important information. Thus, undersampling techniques with heuristics, to select which instances to keep and/or remove were also evaluated (Brownlee, 2020a, p. 140). There exists several techniques which uses heuristics for undersampling, in this thesis the following were examined.

The Tomek Links method creates pairs (links) consisting of instances from opposite classes that lie closest to the borderline, based on the instances with the smallest Euclidian distance. The majority class in each pair (link) is then removed to increase the distance between the classes, which ultimately removes noise and restores balance in the data (Brownlee, 2020a, p. 150).

Another method for detecting noisy instances on the borderline of the data that were explored was Edited Nearest Neighbors (ENN). ENN locates and removes the misclassified instances based on the three nearest neighbors in the data and then applies a classification rule equal to a single nearest neighbor to make decisions (Brownlee, 2020a, p. 152).

Nevertheless, under-sampling techniques alone might be more suitable for less imbalanced data. Consequently, as the data used in this thesis were highly imbalanced, a combination of under- and oversampling techniques was presumed to be more effective (Brownlee, 2020a, p. 117).

4.2.1.3 Combinations of Resampling Techniques

A combination of undersampling and oversampling could be beneficial to increase the model performance by reaping the positive and diminishing the negative effects of both sampling techniques. Another method to handle the imbalance is therefore to create a pipeline which combines both undersampling and oversampling techniques on the training data. The order does not directly matter as the sampling impacts the opposite classes (Brownlee, 2020a, p. 122). The combination of resampling was applied to the data using a pipeline to find the ultimate combination of sampling technique and sampling strategies[17]. The combination of SMOTE and RUS ultimately produced the best result based on the cross-validation score [18] on the training data. However, the improvement was marginal compared to the two techniques individually. Based on these results, it was decided to continue evaluating these two techniques (RUS & SMOTE) individually and in combination with a focus on finding the most optimal sampling strategy.

4.2.2 Cost-Sensitive Learning

As there are four times higher costs associated with Default compared to the potential gain associated with Non-Default[19], cost-sensitive learning was

[17] The amount of data to generate (minority) and remove (majority) based on a given percentage of the class.

[18] Dividing the training data into k folds, training on k-1 folds and testing on the remaining fold. The results were averaged across all testing scores (Provost & Fawcett, 2013, p. 12,127).

[19] The Company states that they on average need 4 Non-Defaulting customers to make up for the loss of 1 Defaulting customer.

explored as an alternate solution to handle the imbalance in the training data. Cost-sensitive learning takes into consideration the costs of each class when training the model, instead of modifying the underlying balance in the data. This could be beneficial, as misclassifying an instance of the minority class (Default) is considered worse than misclassifying an instance of the majority class (Non-Default). To diminish the costs of misclassifications, cost-proportional weighting, could therefore be implemented. The weighting penalize the model more for errors made on the minority class, and less for errors made on the majority class (Brownlee, 2020a, p. 178), (Mumtaz, 2020). When evaluating various weights, it was evident that a training data with no resampling, but higher weights for the minority class ultimately provided the most optimal results.

4.3 Measuring Model Performance

Various performance metrics can be used to evaluate and quantify the performance of predictive models. The result of each evaluation metric is based on different underlying assumptions about what is considered of importance. It is therefore crucial to use the right metric for evaluating models for the classification problem. Additionally, the imbalance in the data excludes some of the standard metrics as, for instance Accuracy[20], as they can be both misleading and unreliable (Brownlee, 2020a, p. 37). The utilized performance metrics are presented below.

4.3.1 Confusion Matrix

A common method for evaluating the performance of a model is to use the confusion matrix. The confusion matrix separates and visualizes the decisions made by the model. This makes it easier to gain a better understanding of how the model confuses one class for another (Provost & Fawcett, 2013, p. 189).

[20] Accuracy is a common method for evaluating the performance of a model. However, with highly imbalanced data this metric is often unreliable (Provost & Fawcett, 2013, p. 189). Due to the imbalance in the data, any naive model which predicts all instances to be Non-Default (part of the majority class) would achieve an accuracy score of $\approx 97\%$. As the goal is to capture the Defaults (minority class), the model should achieve an accurate prediction of both classes.

As can be seen from figure 4.2, the confusion matrix divides the instances into 4 blocks. The True Negative (TN), which represents the instances that are actually negative and that also are correctly predicted as negative. False Negative (FN) represents the instances that are actually positive, but that the model misclassified as negative. False Positives (FP) are the instances that are actually negative but that the model misclassified as positive. True Positive (TP) represents the instances that are actually positive and that also are correctly predicted as positive. Derived from the confusion matrix, there exist several other performance metrics that capture various aspects of the confusion matrix (Provost & Fawcett, 2013, p. 203). For this thesis, the optimal confusion matrix would be a high TP with a low FP.

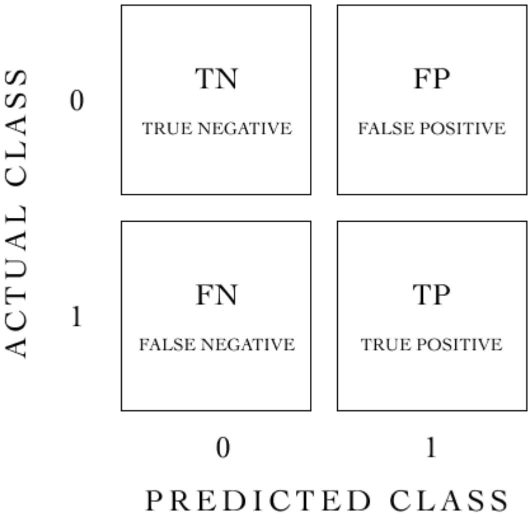


Figure 4.2: Generic confusion matrix

4.3.2 Precision and Recall

As an alternative performance metric, it is possible to use two other metrics based on the confusion matrix; precision and/or recall. Both of these metrics quantify model performance based on the most important class, the minority (Default).

Precision measures the TP rate, which indicates how well the model is at correctly predicting instances of the minority class (Default). However, precision does not take into consideration how much of the majority class (Non-Default) that has been misclassified (FP) to achieve the given level of correctly predicted minorities (Brownlee, 2020a, p. 63, 64).

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

On the other hand, recall measures the number of predicted instances of the minority class (Default) based on all minority instances in the data. Consequently, recall gives a better indication of how well the model correctly predicts Defaults, as recall takes into consideration the number of misclassified Defaults (FN) (Provost & Fawcett, 2013, p. 203, 204).

The metric most appropriate to use depends on the given problem. If the modeler wants to focus on minimizing FP, then precision is the most appropriate. Whereas if the modeler wants to focus on minimizing the FN negatives, then recall is the most appropriate (Brownlee, 2020a, p. 63, 64). Hence, for classifying Defaults, recall is considered more important. As the data is imbalanced, the goal is to improve recall while not extensively lowering the precision. However, this could be challenging as an increase in one of these metrics often comes at the expense of a decrease in the opposite metric (Brownlee, 2020a, p. 63, 64).

4.3.3 F1-Score

To avoid choosing between precision and recall, one could use the F1-score, which provides a score that expresses both precision and recall. Thus, it is also a metric that is commonly used when working with imbalanced data (Provost & Fawcett, 2013, p. 203, 204). However, it should be noted that the F1-score does not take into consideration the correctly predicted majority class (TN), which can make this performance metric misleading (Brownlee, 2020a, p. 64).

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.3}$$

4.3.4 Mattews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) takes all values of the confusion matrix into account and indicates whether there exists correlation between the predicted- and the true class. Hence, a high value could only be achieved if both classes are predicted accurately. Whereas the other metrics go from 0 to 1, the

MCC goes from -1 to 1. Thus, if the value is 0, this indicates that the model is no better than guessing. This metric is not affected by the disproportionately of the classes and is therefore suitable for measuring model performance on imbalanced classes (Chicco et al., 2021). Consequently, this metric was chosen as the most important evaluation metric.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4.4}$$

The MCC score represents the overall performance of both classes and is therefore highly emphasised in the final evaluation. However, this score should not extensively compromise neither recall nor precision.

4.4 Models & Comparison of Performance

To find the ultimate machine learning model for predicting Default, various models were explored and compared. Despite the focus on machine learning in this thesis, there is no need to choose a more advanced model if the same results could be achieved with a more traditional model that is easier to implement and interpret. To justify the use of machine learning models, the result obtained with the simple Linear Probability Model (4.4.1.1) was defined as the baseline. The following traditional models and the machine learning models (with and without optimization) were then compared to the baseline results.

4.4.1 Traditional Statistical Models

The Linear Probability Model is commonly used to study binary classification as it is simple to implement and the results can easily be interpreted. However, due to limitations[21] of the Linear Probability Model, the Logistic Regression (commonly referred to as the Logit model) is often preferred (Sucarrat, 2017, p. 126). As they are both commonly used and easy to interpret, these models have been chosen as the traditional models.

[21] First, due to the error term, the model is strictly not compatible with a Y variable equal to 0 and 1. Second, the model does not guarantee a $Pr(Y=1|\mathbf{X}) \in [0,1]$ (Sucarrat, 2017).

The simple regressions were trained only on the variable NOT IN DR, as it represents whether or not an applicant has any prior Consumer Debt. This variable was chosen for the simple models as an applicant with no prior Consumer Debt was assumed to be less likely to Default. Furthermore, the classifications were conducted with thresholds equal to the sample mean of y ($\overline{y}$) and is equal to 0.5. With a threshold equal to 0.5, the models did not predict any of the Defaults accurately, which is not surprising as the Default instances only account for 2.7% of the instances in the data. Nevertheless, as machine learning models utilize the threshold (0.5) as the standard value, the threshold was included to create a basis of comparison.

4.4.1.1 The Linear Probability Model

The model is given by

$$Y = B_0 + B_1 X_1 + ... + B_k X_k + u \tag{4.5}$$

$$Pr(Y = 1|\mathbf{X}) = B_0 + B_1 X_1 + ... + B_k X_k \tag{4.6}$$

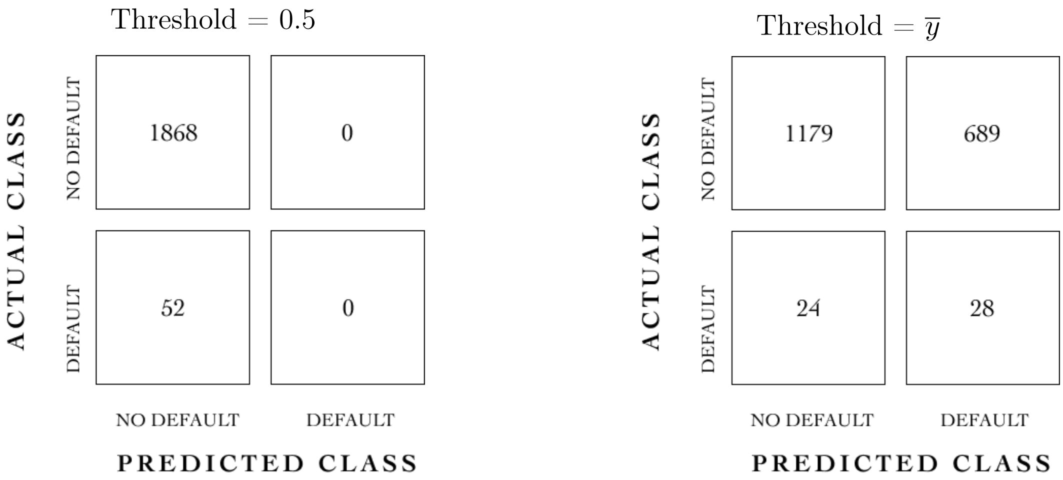
In this thesis, a simple Linear Probability Model to predict Default, based on whether or not the customer exists in the Debt Registry[22], is given by

$$\widehat{Pr}(Y = 1|X) = 0.025 + 0.006 \times NOT\,IN\,DR \tag{4.7}$$

The model gives a predicted likelihood of 2.5% of an applicant who is not represented in the Debt Registry to Default, while an applicant who is represented in the Debt Registry has a predicted probability of Default equal to 3.1%. The provided probabilities can with a set threshold, be used to classify applicants. The threshold determines the cut-off for deciding whether a prediction is classified as 0 or 1. A commonly used threshold is 0.5, which translates to any observation with a $Pr(Y_i = 1|X_i) \geq 50\%$ is classified as 1. In this case, a threshold equal to 0.5 would result in all observations classified as 0, which means that there are no observations which are more than 50% likely

[22] If an applicant have no prior Consumer Debt they will not be listed in the Debt Registry. This binary variable is therefore an indicator of whether or not the applicant has prior debt.

to be Default based on the explanatory variable. To shift the model towards detecting Default, the threshold was set to the sample mean of y ($\overline{y}$). At this threshold, the model was able to predict $\approx 54\%$ of all Defaults. However the model also misclassified $\approx 37\%$ of the applicants who would not Default.



Threshold $= 0.5$

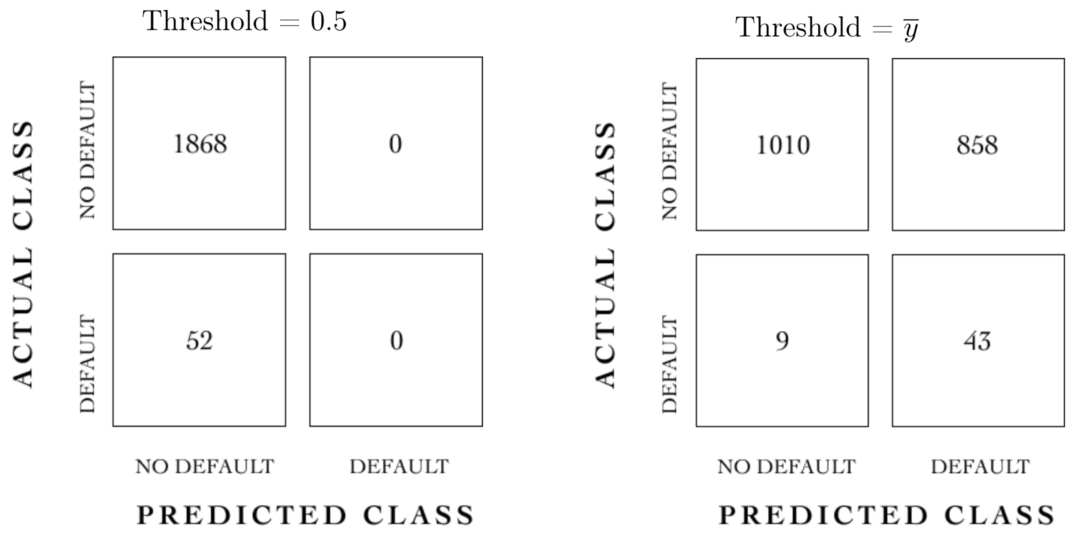
ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1868	0
DEFAULT	52	0

Threshold $= \overline{y}$

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1179	689
DEFAULT	24	28

Figure 4.3: Simple Linear probability model

By expanding the model to a Multiple Linear Probability Model, which includes all available data[23], the model (with a threshold equal to the sample mean of y ($\overline{y}$)), is able to predict $\approx 83\%$ of all Defaults. However, this comes at the expense of a $\approx 46\%$ misclassification of all Non-Defaults. With the results of the linear probability model as a baseline, the goal is to find a model which more accurately predicts Defaults.



Threshold $= 0.5$

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1868	0
DEFAULT	52	0

Threshold $= \overline{y}$

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1010	858
DEFAULT	9	43

Figure 4.4: Multiple Linear probability model

[23] As the categorical variables were dummy-encoded, for all categorical variables with k values onlyk-1 dummy variables were included to avoid the dummy-trap (Sucarrat, 2017, p. 106).

4.4.1.2 Simple Logistic Regression

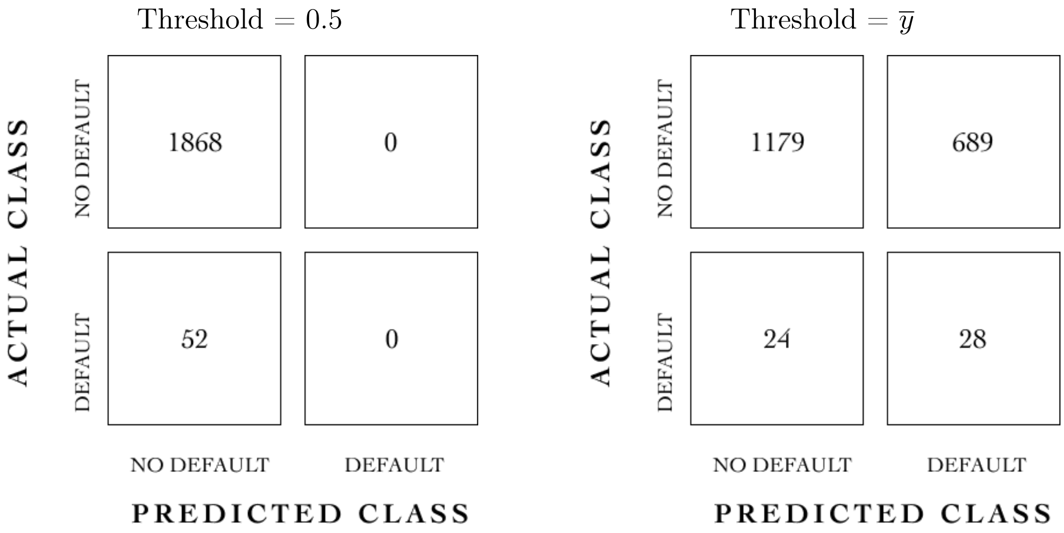
The Logistic regression is given by

$$Pr(Y=1|X)=\frac{e^{\mathrm{L}}}{1+e^{\mathrm{L}}} \qquad L=B_0+B_1X \tag{4.8}$$

where L is the natural logarithm of the relationship between the 0 and 1 probabilities (Sucarrat, 2017, p 126). Ultimately, the model predicts each instance's class probability and returns a value $\in (0,1)$ which is the estimated probability of Default for each observation (Provost & Fawcett, 2013, p. 96). By estimating a simple Logit model based on whether or not an applicant is present in the Debt Registry, the models performance can be compared to the Simple Linear Probability Model.

$$\widehat{Pr}(Y=1|X)=\frac{e^{\hat{L}}}{1+e^{\hat{L}}} \qquad \widehat{L}=-3.662+0.211\times NOT\,IN\,DR \tag{4.9}$$

The predicted probability of Default for an applicant who is registered in the Debt Registry, is 0.31% and for the ones who are not, the predicted probability is 0.25%. The confusion matrix (based on a threshold = 0.5) displays that the model classified all observations as Non-Default, exactly like the Simple Linear Probability Model. However, with a threshold equal to the sample mean of y ($\overline{y}$), the model obtains the exact same results as the Linear Probability Model (figure 4.3).



Threshold = 0.5

ACTUAL CLASS	PREDICTED CLASS: NO DEFAULT	PREDICTED CLASS: DEFAULT
NO DEFAULT	1868	0
DEFAULT	52	0

Threshold = $\overline{y}$

ACTUAL CLASS	PREDICTED CLASS: NO DEFAULT	PREDICTED CLASS: DEFAULT
NO DEFAULT	1179	689
DEFAULT	24	28

Figure 4.5: Simple Logistic regression model

As the goal is to predict Default it is evident that changing from the Linear Probability Model to a Simple Logistic Regression model (which is more advanced) did not provide better results. Thus, various machine learning models will be explored in the following section. However, one of the main disadvantages of machine learning models compared to the Logistic Regression model is that they have lower interpretability because they operate more like a black box. Thus, these models might require more time spent on optimizing the hyperparameters, and interpreting the underlying results (Kho, 2018).

4.4.2 Machine Learning Models

Initially, 6 of the most commonly used machine learning models across 5 different types[24], were considered (Brownlee, 2019). The models were scored based on the average cross-validation score[25] on the training data. The three machine learning models with the highest average cross-validation score (table 4.1) were further compared to the baseline models. It should be noted, that cross-validation was only used on the training data to compare the 6 machine learning models. The models' predictive performance were (as the traditional

[24] The algorithms type refers to their functionality and are grouped with other models based on similar functionality. For instance, Random Forest and XGBoost are both decision tree models based on an ensemble of multiple trees (Brownlee, 2019).

[25] Cross validation is conducted by dividing the training data into k folds, training on k-1 folds and testing on the remaining fold. Then the results were averaged across all test scores (Provost & Fawcett, 2013, p. 126,127).

statistical models) obtained with the holdout strategy.

Type	**Model**	**Average CV Score**
Decision Tree	RandomForest	0.82
Decision Tree	XGBoost	0.81
Regression	Logistic Regression	0.71
Cluster	K-NearestNeighbors	0.69
Instance based	Support Vector Machine	0.21
Bayesian	Gaussian Naive Bayes	0.21

Table 4.1: Average cross validation score for model selection

4.4.2.1 Machine Learning with Logistic Regression

Logistic regression is a statistical tool commonly used for classification and has therefore been adopted by the machine learning field (Brownlee, 2020c). However, the model is not well suited for predicting on imbalanced data without adding weights to the classes (Brownlee, 2020a, p. 193). The Multiple Logistic Regression was therefore included as part of the machine learning section, as weights can be passed as a parameter before training the model.

The Multiple Logistic Regression was developed with the use of the machine learning library scikit-learn (Pedregosa et al., 2011a). When predicting binary labels with this library, the standard threshold is equal to 0.5. However, as the Simple Logistic Regression obtained superior results with a threshold equal to the sample mean of y ($\overline{y}$), the predicted probabilities were used to create classifications with a threshold equal to the sample mean of y ($\overline{y}$) for the Multiple Logistic Regression.

The multiple model is an extension of the simple model where more than one explanatory variable is included (Sucarrat, 2017). The categorical variables were previously dummy-encoded, and for all categorical variables with k values, only k-1 dummy variables were included to avoid the dummy-trap[26]. All other

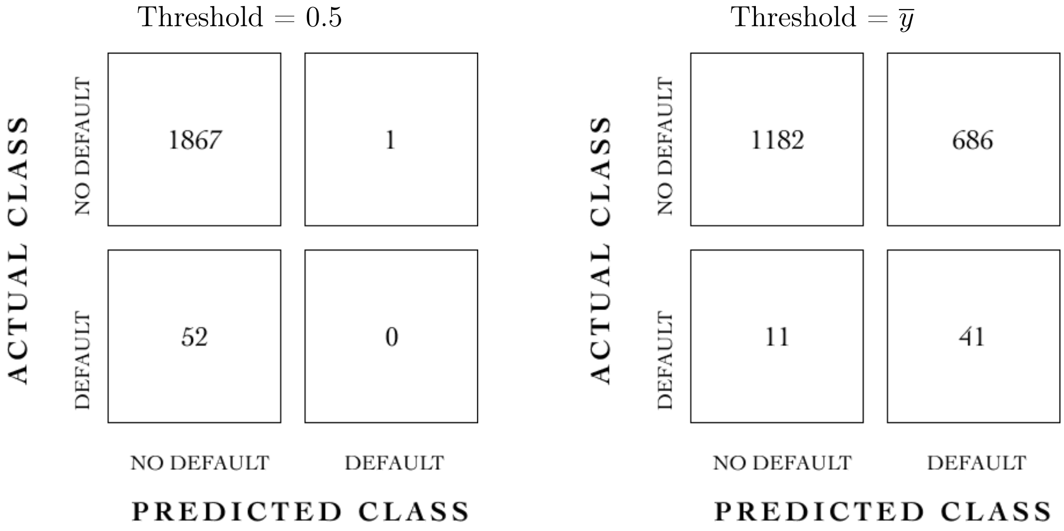
[26] Multicollinarity can occur if one binary variable can be written as an exact linear combination of other binary variables (Sucarrat, 2017, p. 106).

variables in the data were included. The Multiple Logistic Regression model is given by

$$Pr(Y = 1|\mathbf{X}) = \frac{e^{\mathrm{L}}}{1 + e^{\mathrm{L}}} \qquad\qquad L = B_0 + B_1X + ...B_kX_k \tag{4.10}$$

This classifier is a highly used model with multiple advantages. Compared to other machine learning models, the Logistic Regression model is easier to interpret as it is possible to assess which of the variables that has the highest impact on the predicted value. Second, the model is both simple and fast, which makes it easy to use for both new predictions and future maintenance (Keboola, 2020).

The machine learning version of Logistic Regression was deployed with the use of scikit-learn and its standard parameter values (Pedregosa et al., 2011a). As can be seen from the confusion matrix based on a threshold equal to 0.5, no Defaults (TP) are predicted correctly.



Threshold = 0.5

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1867	1
DEFAULT	52	0

Threshold = $\overline{y}$

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1182	686
DEFAULT	11	41

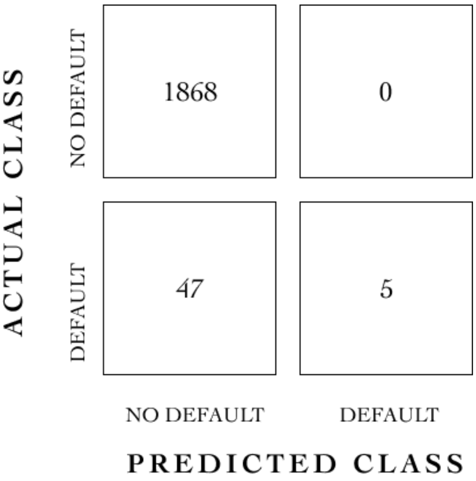
Figure 4.6: Multiple Logistic regression model

The model predicts 1 instance as Default, however, this instance is a misclassification (FP). With the standard parameter values, the results are not improved compared to the more traditional models. To compare the model against the better performing traditional models, the threshold was adjusted to the sample mean of y ($\overline{y}$). With a lower threshold, the predictive performance was drastically improved and the Logistic Regression outperformed both the Simple Logistic Regression, the Simple- and the Multiple Linear Probability Model. Despite the increase in prediction of Defaulting Applicants, the model still misclassifies several of the Non-Defaulting Applicants.

4.4.2.2 Random Forest

The Random Forest classifier[27] is a bagged decision tree. Decision tree algorithms take into consideration the variables in the data, and based on the variables, split the data into subsections until the model cannot split further. The term bagging means that the model contains multiple decision trees, which are then trained on multiple subsets of the training data before the final predictions are averaged. Random Forest further improves the bagging technique by decorrelating the trees through random splits on a smaller subset of variables in the data. The subsets results in quicker training speeds than other decision trees, which makes it possible to work with more variables.

On the other hand, if the given data consists of several strong predictors, then the trees could be highly correlated as the decision trees would be quite similar. This model needs less preprocessing and transformation compared to other models. Compared to boosting models which run sequentially, the advantage of bagging is that they could be run in parallel, thus resulting in faster computing time (Kho, 2018). In comparison to Logistic Regression, Random Forest is more robust to outliers and nonlinear data (Kho, 2018).



	PREDICTED CLASS	
ACTUAL CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1868	0
DEFAULT	47	5

The Random Forest accurately predicts 19% more of the Defaults (TP) compared

[27] The model is retrieved from the scikit-learn library for machine learning (Pedregosa et al., 2011b).

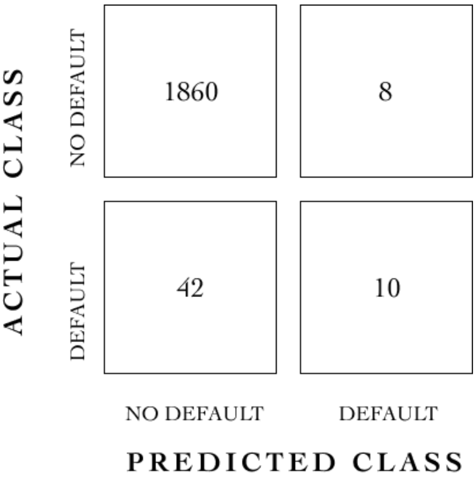
to the Logistic Regression with baseline values for both models.

Additionally, Random Forest does not misclassify any Non-Defaulting applicants (FP) whereas the Logistic Regression misclassified 1. The Logistic Regression with a threshold equal to the sample mean of y ($\overline{y}$) accurately predicts 79% of the Defaulting Applicants, but the model also misclassifies 37% of the Non-Defaulting Applicants.

4.4.2.3 eXtreme Gradient Boosting

The classifier eXtreme Gradient Boosting[28] (XGBoost) is like Random Forest part of the tree-based ensemble algorithms. Whereas Random Forest uses bagging, where the model learns in parallel, XGBoost utilizes boosting where the model learns sequentially from previous iterations. The model aims to improve the predecessor predictive performance and correct errors (Nikulski, 2020). Furthermore, XGBoost has many opportunities for optimizing the hyperparameters, which can lead to a better performing model, but also requires more expertise and time spent on tuning the model compared to Random Forest (Nikulski, 2020).

XGBoost with standard parameters and a threshold equal to 0.5, correctly predicts 19% of the Defaults (TP) and only misclassifies 0.4% of the Non-Defaults (FP). In comparison, the Logistic Regression with a threshold equal to the sample mean of y ($\overline{y}$), correctly predicts 4 times more of the Defaulting Applicants. However, the Logistic Regression also misclassifies 86 times more Non-Defaulting applicants.



		PREDICTED CLASS	
		NO DEFAULT	DEFAULT
ACTUAL CLASS	NO DEFAULT	1860	8
	DEFAULT	42	10

[28] The model is retrieved from the Python API for XGBoost (XGBoost Python Package, n.d.)

4.4.3 Optimization of Machine Learning Models

The following section aims to optimize each of the machine learning models as the models predictive performance is affected by the input data and the values of the parameters. It is presumed that optimizing the models could maximize the models predictive performance. Initially only the hyperparameters were optimized. However, as this resulted in suboptimal results, RandomSearch was utilized to find the best combination of hyperparameters, resampling and variable selection for each model (section 4.4.3.3).

4.4.3.1 Hyperparameters

Hyperparameters are the input parameters for a machine learning algorithm that can be explicitly set before the training process. The values of these parameters determine how well the model will perform, and should therefore be set at an optimal level (Badr, 2019a). This process is also referred to as parametric modeling and aims to find the optimal values that fits the model to the training data (Provost & Fawcett, 2013, p. 81).

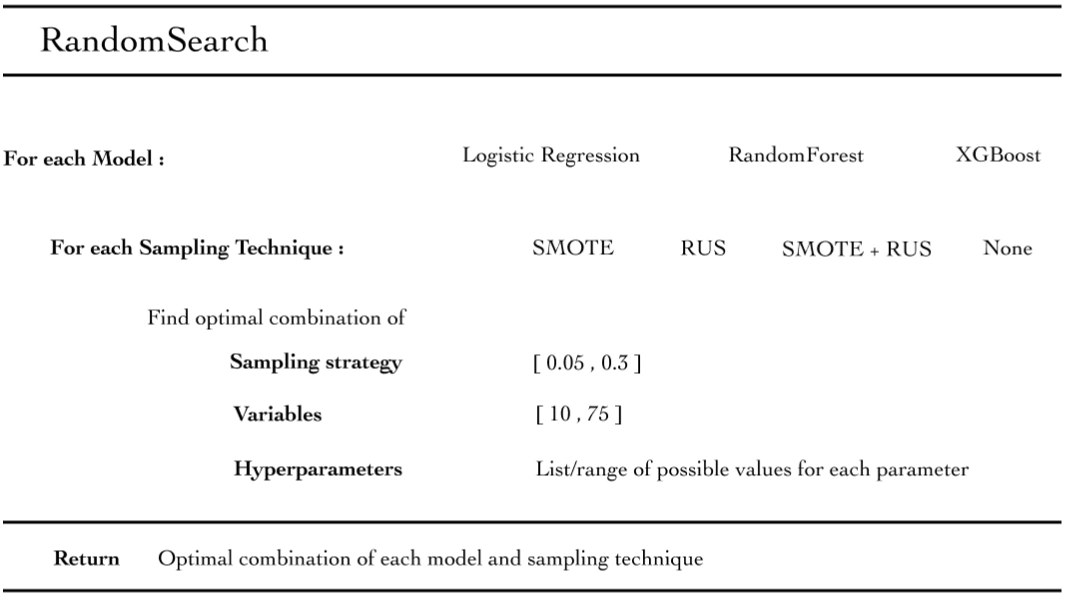
To streamline this process, there are several automated tools available such as GridSearch and RandomSearch. These tools automatically tune the hyperparameters to create the best combination for a model to maximise the performance. Both GridSearch and RandomSearch were considered as automated tools to tune the hyperparameters. However, only RandomSearch was applied. The reason for not choosing GridSearch is the high dimensionality of the hyperparameters in the various models. This method combines all possible combinations which can be highly time consuming as the number of evaluations required to find the optimal solution grows exponentially. RandomSearch on the other hand, takes in the grid of parameters and possible values and conducts n iterations where random combinations of the parameters are explored to find the best combination (Senapati, 2018).

4.4.3.2 Variable Selection

After the data preparation, the data is purely numeric and understandable for a machine learning algorithm. However, the result of this transformation is a large increase in variables which may be subject to the issue of dimensionality[29]. In order to avoid this problem, the number of variables were reduced with the use of an automated tool from scikit-learn for variable selection (Pedregosa et al., 2011a). This tool is a univariate variable selection method that is based on an F-test, and estimates the degree of linear dependency between two random variables (scikit-learn 0.24.2, 2021). By computing the ANOVA-F[30] value for the training set, it returns the K best scoring variables which is then used in the training.

4.4.3.3 RandomSearch

In the process of finding the best variables, resampling technique, and hyperparameters, it became clear that these three dimensions greatly affected each other. By isolating- and optimizing them individually, the results were suboptimal and they were therefore combined in a pipeline to find the best combination.



```
RandomSearch

For each Model :                                  Logistic Regression    RandomForest    XGBoost

    For each Sampling Technique :                 SMOTE    RUS    SMOTE + RUS    None

        Find optimal combination of
            Sampling strategy                     [ 0.05 , 0.3 ]
            Variables                             [ 10 , 75 ]
            Hyperparameters                       List/range of possible values for each parameter

    Return    Optimal combination of each model and sampling technique
```

Figure 4.7: Pipeline for finding optimal combinations using RandomSearch

Four versions of the data were created based on different resampling techniques (RUS, SMOTE, SMOTE+RUS and None). Then, four pipelines were

[29] When an algorithm is presented with too many variables (dimensions) observations may appear equidistant from all other, making it difficult to create meaningful clusters (Yiu, 2019).

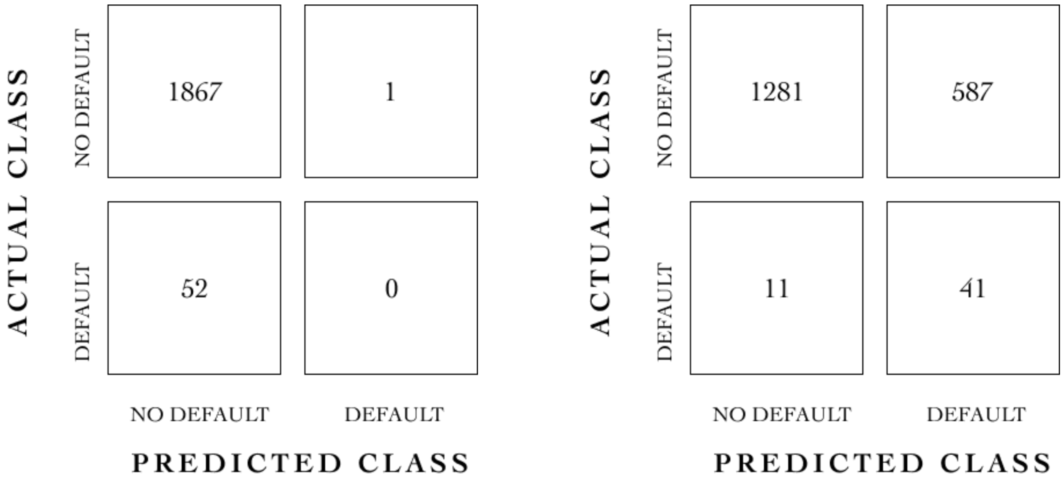
[30] The ANOVA procedure is an analysis of variance that is used to compare means within and amongst groups to confirm or deny that the means are equal (Sullivan, n.d).

executed for each of the three models were variables and hyperparameters were optimized to the different versions of the data. The grid of the hyperparameters and the range for variable selection remained constant over the four variations of data[31]. A general description of each pipeline is shown in figure 4.7. For the respective grid values see appendix A2.3. The results of the pipeline revealed that the most optimal performance was achieved with no resampling, and a minority class weighted higher than the majority. In the following section the best combination of each model will be evaluated.

4.4.3.4 Optimized Logistic Regression

All results are obtained with a threshold equal to 0.5.

Baseline Threshold=0.5 | Optimized Threshold=0.5



Baseline, Threshold=0.5:

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1867	1
DEFAULT	52	0

Optimized, Threshold=0.5:

ACTUAL CLASS \ PREDICTED CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1281	587
DEFAULT	11	41

MCC = *n/a* | MCC=0.14

Recall = *n/a* | Recall=0.79

Precision = *n/a* | Precision=0.07

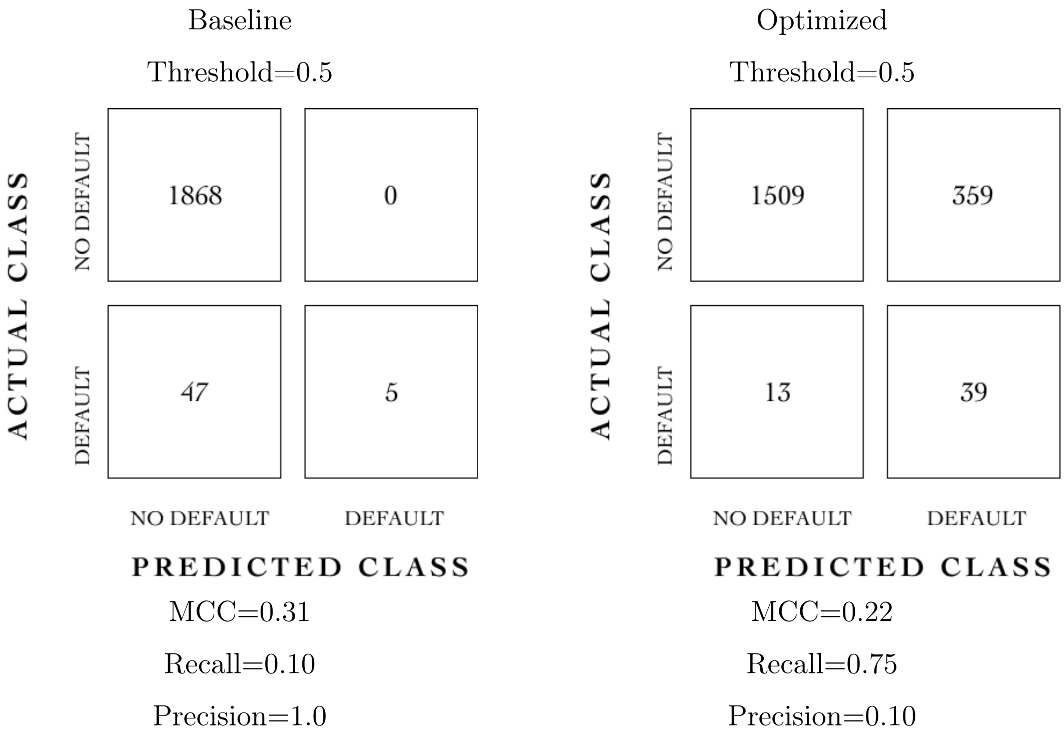
Figure 4.8: Logistic Regression

For the baseline model the chosen metrics are not applicable[32] as the model does not correctly predict any Defaults. Since, the Optimized Model correctly predicts 79% (recall) of all Defaults, the Optimized model outperform the baseline. However, the MCC score is relatively low, 0.14, which indicates that the majority class is extensively misclassified.

[31] with the exception of SMOTE+RUS where the sampling strategy grid refers to SMOTE, and the RUS had a range of 0.2 - 0.5.

[32] This would results in a zero-division.

4.4.3.5 Optimized Random Forest



Baseline

Threshold=0.5

ACTUAL CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1868	0
DEFAULT	47	5

PREDICTED CLASS

MCC=0.31

Recall=0.10

Precision=1.0

Optimized

Threshold=0.5

ACTUAL CLASS	NO DEFAULT	DEFAULT
NO DEFAULT	1509	359
DEFAULT	13	39

PREDICTED CLASS

MCC=0.22

Recall=0.75

Precision=0.10

Figure 4.9: Random Forest

By examining the recall it is evident that the Optimized model predicts more actual Defaults than the baseline. Interestingly enough, the MCC score is higher for the baseline model as it predicts the Non-Defaults perfectly. Furthermore, the F1-score remains unchanged, which is a good example of how the score can be misleading. Furthermore, the Optimized Random Forest misclassifies 2 more actual Defaults compared to the Optimized Logistic Regression. However, the Optimized Random Forest ultimately outperform the Optimized Logistic Regression as it receives a higher MCC score due to a more correct prediction of the majority class, and a high recall due to correct prediction of the minority class.

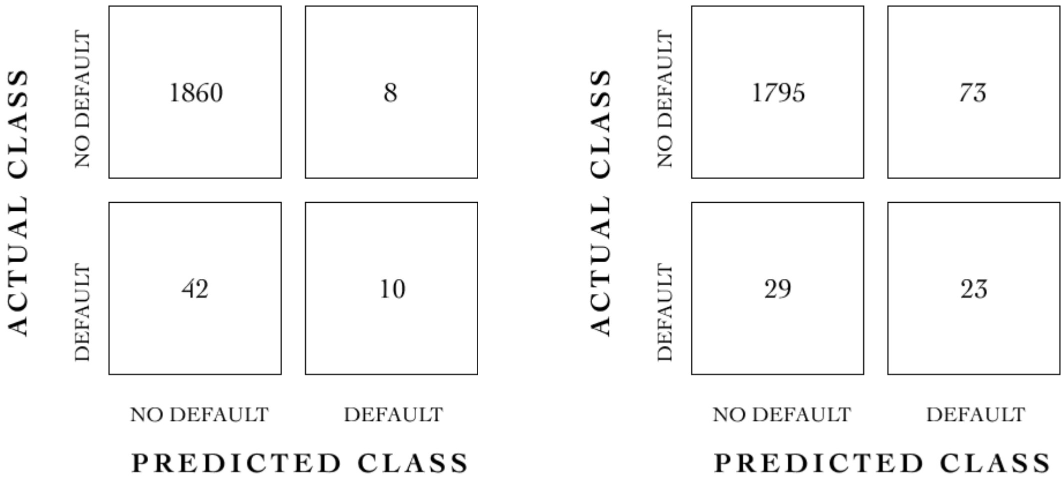
4.4.3.6 Optimized XGBoost

Baseline

Threshold=0.5

Optimized

Threshold=0.5



ACTUAL CLASS	NO DEFAULT (predicted)	DEFAULT (predicted)
NO DEFAULT	1860	8
DEFAULT	42	10

ACTUAL CLASS	NO DEFAULT (predicted)	DEFAULT (predicted)
NO DEFAULT	1795	73
DEFAULT	29	23

MCC=0.32

Recall=0.19

Precision=0.56

MCC=0.30

Recall=0.44

Precision=0.24

Figure 4.10: XGBoost

As the goal were to achieve a high TP, while simultaneously achieve a low FP, the Optimized XGBoost were considered as the best performing model for this classification problem. Ultimately, the Optimized XGBoost were the model with the highest MCC score without compromising the precision, the recall and the amount of True Positives (TP).

4.5 Result of Model Comparison

All results obtained from the model comparison are displayed in table 4.2. These results are obtained from classifications made on the holdout data. As can be seen from the results displayed in the table, the optimized machine learning models ultimately provided the best results with the exception of Random Forest with baseline values. The model achieves a precision equal to 1.00 as no Non-Defaults are misclassified (FP=0). However, the model only captures 5 of the Defaults and is therefore considered to perform suboptimal as the goal is to predict Default.

Model	**Precision**	**Recall**	**F1**	**MCC**	**Average**
Traditional					
Threshold = 0.5					
Simple Linear Probability	-	-	-	-	-
Multiple Linear Probability	-	-	-	-	-
Simple Logistic Regression	-	-	-	-	-
Threshold $= \overline{y}$					
Simple Linear Probability	0.04	0.54	0.07	0.04	0.17
Multiple Linear Probability	0.05	0.83	0.09	0.12	0.27
Simple Logistic Regression	0.04	0.54	0.07	0.04	0.17
Multiple Logistic Regression	0.06	0.79	0.11	0.14	0.28
Machine Learning					
Threshold = 0.5					
Baseline Values					
ML Logistic Regression	-	-	-	-	-
RandomForest	1.00	0.10	0.18	0.31	0.40
XGBoost	0.56	0.19	0.29	0.32	0.34
Optimized Values					
Logistic Regression	0.07	0.79	0.13	0.14	0.28
RandomForest	0.10	0.75	0.18	0.22	0.31
XGBoost	**0.24**	**0.44**	**0.31**	**0.30**	**0.32**

Table 4.2: Performance of all models

Amongst the optimized machine learning models, the best performing models were Random forest and XGBoost. The Random Forest has a high recall, which illustrates that the model correctly predicts a large number of Defaults. However, as the model misclassifies a large amount of Non-Defaults, the precision is quite low, which lowers the MCC-score.

The XGBoost achieves a higher MCC score and precision compared to the Random Forest. On the other hand, the recall is lowered, as fewer Defaults are predicted correctly. Nevertheless, XGBoost achieved the highest MCC score without extensively compromising recall and presicion. The XGboost additionally achieves the highest average score [33] amongst the optimized machine learning models.

Consequently, XGBoost was considered the best performing model compared to both traditional- and other machine learning models. The grid of hyperparameters used to obtain these results can be seen in table A2.4. Further results obtained with the model are discussed in Chapter 5.

[33] The sum of all scores obtained for the model divided by the number of metrics.

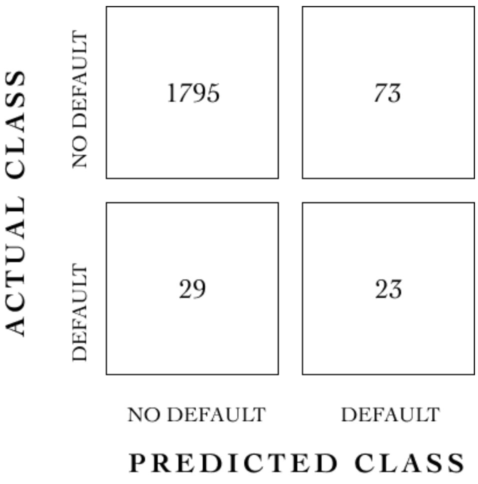
5 Detailed Examination of XGBoost

This chapter further examines the results obtained with the best performing model from the previous chapter, XGBoost, when deploying the model on the holdout. The prediction results are assessed and discussed in more detail, as well as the consequences of shifting the threshold. Furthermore, the importance of the variables are compared and the potential monetary gain, as well as the impact of the Debt Registry on the predictive performance are assessed. Whereas, the previous chapter, Chapter 4, provided the grounds for choosing the predictive model. The examination in this chapter provides the foundation for the answers to the research question in the next chapter, Chapter 6.

5.1 Prediction Results

The model was trained with the best 66 variables on 85 % of the data and then tested on the remaining 15%. The model's predictive performance achieved an MCC score equal to 0.3, where the maximum correlation score is 1.0. As the MCC score takes a value between [-1,1], a score of 0.3 might be perceived as low. However, as the model is built on data with a high degree of entropy and noise due to human irrationality, a result above 0.2 could be considered satisfactory [34].

Presuming that the Company deployed the model on the following 1920 applicants in the test data, 5% (96) of the applicants would have been declined and the remaining 95% (1824) of the applicants would have been granted credit.



		PREDICTED CLASS	
		NO DEFAULT	DEFAULT
ACTUAL CLASS	NO DEFAULT	1795	73
	DEFAULT	29	23

Figure 5.1: Predicted results

Hence, the Company would decline 73 Non-Defaulting Applicants, while they grant credit to 28 Defaulting Applicants. However, if deployed, the misclassified Defaulting Applicants would be unknown

[34] The Company states that they consider a model with an MCC score above 0.2 as high performing.

until the time of default, and the misclassified Non-Defaulting Applicants would never be known as they would have been declined. The model is not able to correctly classify all instances. Nevertheless, the model could be considered an improvement compared to the current process where all (52) of the Defaulting Applicants were accepted.

5.2 Classification Threshold

The model was executed with a standard threshold equal to 0.5, which means that any observation with a predicted probability $\geq 50\%$ was classified as Default. Shifting the threshold is one method to account for the different costs associated with Default and Non-Default (Brownlee, 2020b).

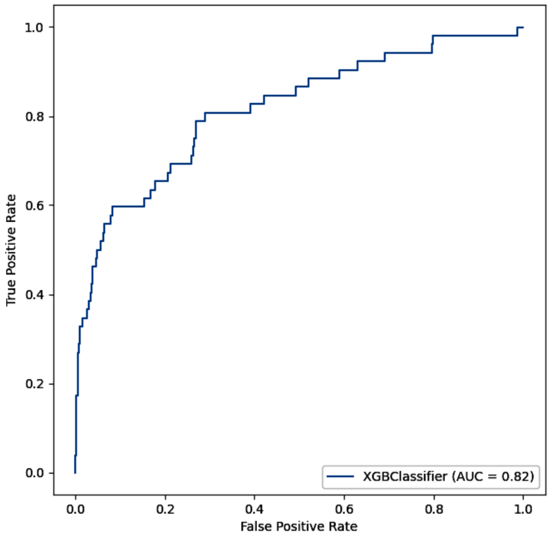


Figure 5.2: AUROC curve

The Area under the Receiver Operator Characteristic Curve (AUROC curve) is a performance metric that is useful for comparing and evaluating thresholds. The AUROC curve indicates how the threshold affects the classification as it ultimately decides which observations should be labeled as Default. It is compiled of two components, the Receiver Operator Characteristics (ROC) which represents the probability curve, and the Area Under The Curve (AUC) which represents the degree of separability. In other words, the ROC curve gives a better evaluation of the true positive rate and the false positive rate through visualization, and the AUC indicates how well the model is able to correctly distinguish and predict the classes (Provost & Fawcett, 2013, p. 219), (Narkhede, 2018).

A model with AUC equal to 0.5 predicts the target as well as a coin toss, which means that the model is solely randomly guessing. The XGBoost model achieves an AUC score equal to 0.82, which indicates a predictive power substantially higher than random guesses.

Metric	**0.4**	**0.5**	**0.6**
MCC	0.22	**0.30**	0.28
Precision	0.13	**0.24**	0.25
Recall	0.50	**0.44**	0.37
F1-Score	0.21	**0.31**	0.30
Average score	0.27	**0.32**	0.30

Table 5.1: The effect of shifting the threshold.

The threshold of the final model was 0.5. If the threshold of the final model were moved either down to 0.4 or up to 0.6, the overall predictive performance would be lowered (table 5.1). Furthermore, if the threshold were to be changed, the model (with the new threshold) should be tested on an extra holdout data to ensure that the model is not overfitted to the test data. Since the trade-off between predicted Default and misclassifications provided a sufficient result, there were no changes made regarding changing the threshold of the model. Thus, every observation with a probability of Default ≥ 0.5 was classified as Default.

5.3 Variable Importance

The variables of the final model were evaluated with the use of a built-in function from scikit-Learn[35]. The 10 most important variables for the classification model are displayed in table 5.2 below. All variables can be seen in figure 5.3.

[35] The variable importance is calculated from the **Gain**, which is the relative contribution of each variable to the model(XGBoost Python Package, n.d.). As XGBoost is an ensemble of trees, the Gain is calculated by estimating each variable's contribution to each tree, based on the level of entropy in the target variable after each split. The total information Gain for each variable is calculated by taking the average reduction in entropy for all trees where the variable is used to split (Lutes, 2019). The variables with the highest Gain-value are considered the most important for generating predictions (Abu-Rmileh, 2019).

5.3.1 Top 10 Variables

Rank	Variable
1.	Habitation Type Homeowner
2.	Marital Status Single
3.	Number of self-registered credit cards
4.	Wealth Category > 1.000.000
5.	NC Product Møre Master Card
6.	Wealth Category 250. - 500.000
7.	Wealth Category 100. - 250.000
8.	Estimated Mortgages Expenses
9.	Total Deposit Alliance
10.	Wealth Category 1. - 50.000

Table 5.2: Most important variables for classification

As can be seen, the most important variable in the model was whether or not the applicant was a homeowner. One could argue that this variable is important for many reasons. For instance, a homeowner probably has a mortgage that can be expanded to cover any future unforeseen expenses, which makes it less likely that these persons will resort to additional credit to cover large expenses. Additionally, one could assume that a homeowner potentially has a more stable economy than a person who is a renter, which makes them less likely to default on their debt. Furthermore, the most important marital status was *single* which arguably is logical based on the assumption that single people are the sole provider in their household, making them more vulnerable to sudden economical changes.

It is important to remember that table 5.2 displays which variables had the highest impact on the model during classification, but the plot does not indicate in which direction the variables shifted during the classification. For instance, it can be assumed that an applicant who is a homeowner will most likely not default, and that a single applicant is more likely to default. Furthermore, a person who has over 1 million NOK in wealth is likely to not default, but a person with wealth under 50,000 NOK is more likely to default. This is merely our interpretation of the plot based on research, as the importance plot itself gives no indication in which direction the variables pull the classification. An interesting observation is that even though the model was provided with all available information from the Debt Registry, none of these variables ended up on the list of the top 10 most important variables. However, the 11th most important variable was *Not in DR* which illustrates that whether or not a person is present in the Debt Registry is more important than the debt information itself.

5.3.2 All Variables

All variables with their respective level of importance are displayed in figure 5.3 below. As previously mentioned, the focus of the importance plot is not on the specific values, but rather the relative difference between the variables.

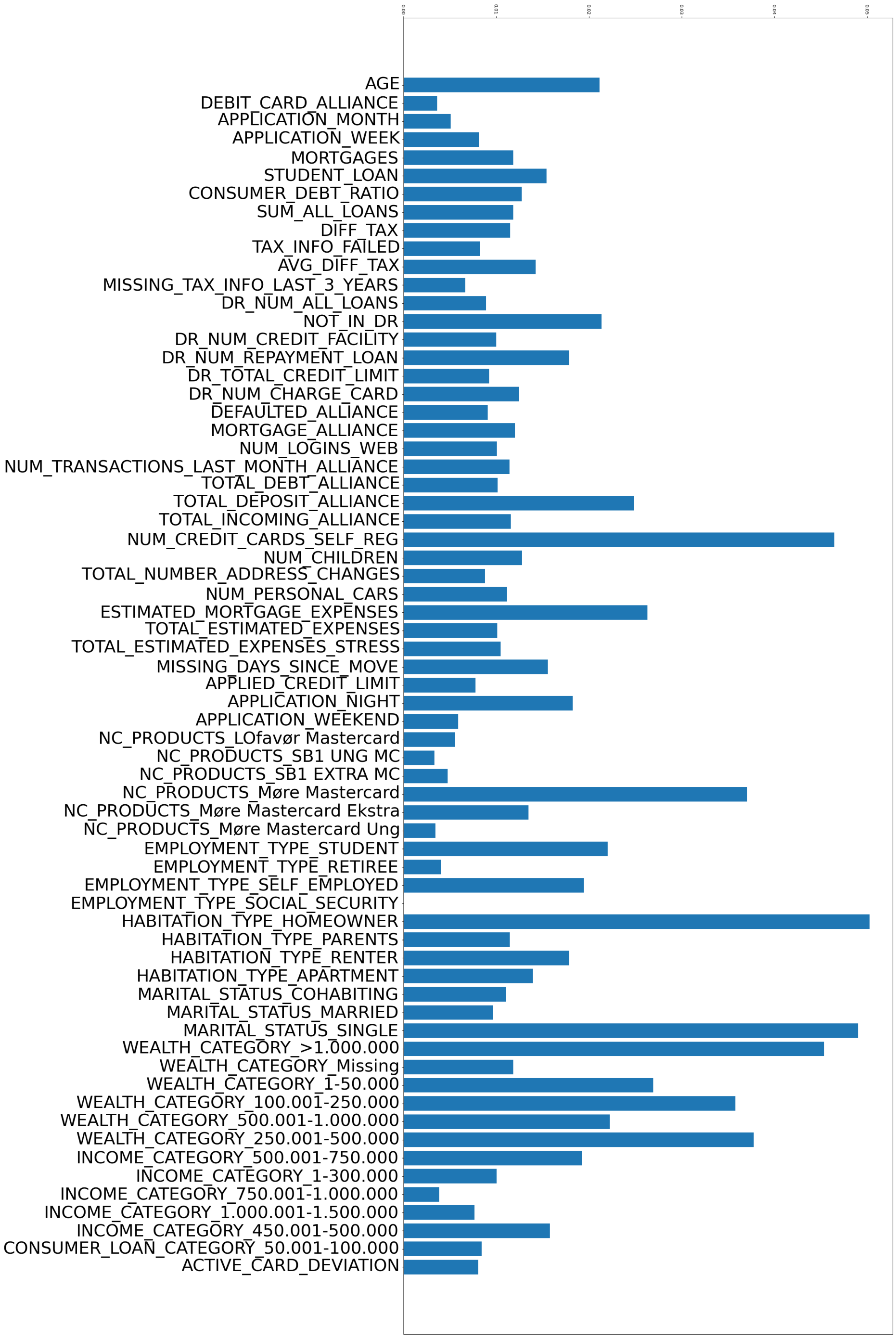


Figure 5.3: Variable Importance Plot Optimized XGB

5.4 Potential Monetary Gain

The potential monetary gain and the impact of the Debt Registry on the model performance are further evaluated based on the results obtained with XGBoost.

Scenario	Realized Loss	Avoided Loss	Realized Income	Missed Income
Without The Model	208,000	-	1,868,000	-
With The Model	116,000	92,000	1,795,000	73,000

Table 5.3: Cost Matrix: Potential Monetary Gain

The current process is represented in the scenario labeled Without The Model, where all applicants were granted credit and 2,7% (345) of the applicants Defaulted within the next 12 months.

The model correctly classifies 96% of the Non-Defaulting Applicants, and 44% of the Defaulting Applicants, which are represented in the scenario labeled With The Model. To encompass all applicants from the Application Period, the results obtained with the test data (15%) are extrapolated by multiplying the results with $\frac{1}{0.15}$. If the model had been implemented before the Application Period, the Company could have avoided a loss of $\approx 613,300$ (the first number in equation 5.2). However, the 4% of the Non-Defaulting applicants which are misclassified, would for the Application Period have resulted in a loss of income $= 486,600$ (the second number in equation 5.2). The total potential saving for the Application Period, is therefore $\approx 126,700$ (the result of equation 5.2). Presumed that the Application Period is representative for an entire year, the Company would potentially have saved $\approx \frac{126{,}700}{4} \times 12 = 380,100$ yearly (the result of equation 5.3).

$$\textbf{Avoided Loss} - \textbf{Missed Income} = \textbf{Gain}$$

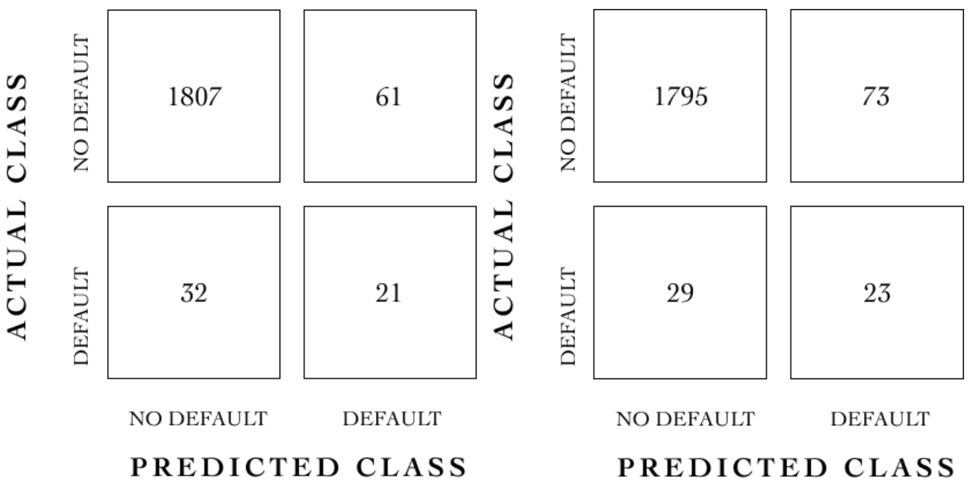
$$92,000 - 73,00 = 19,000 \tag{5.1}$$

$$613,300 - 486,600 = 126,700 \tag{5.2}$$

$$1,839,900 - 1,459,800 = 380,100 \tag{5.3}$$

5.5 Impact of Debt Registry

The impacts of the Debt Registry on the model's predictive performance were measured by training and testing the model without available information from the Debt Registry.


ACTUAL CLASS
PREDICTED CLASS
ACTUAL CLASS
PREDICTED CLASS

(a) Without Debt Registry

	NO DEFAULT	DEFAULT
NO DEFAULT	1807	61
DEFAULT	32	21

MCC=0.30

Recall=0.40

Precision=0.26

(b) With Debt Registry

	NO DEFAULT	DEFAULT
NO DEFAULT	1795	73
DEFAULT	29	23

MCC=0.30

Recall=0.44

Precision=0.24

When comparing the confusion matrixes, it is evident that the precision is slightly improved. However, this comes at the expense of a lowered recall. The MCC remains unchanged, nevertheless, as displayed in figure (b), the model is able to correctly classify 3 more Defaults (TP) with information from the Debt Registry. The model does, however, also misclassify 13 more Non-Defaulting Applicants as Defaults (FP).

According to estimates from the Company, on average they earn 1,000 NOK from a Non-Defaulting Applicant, compared to a loss of 4,000 NOK when accepting a Defaulting Applicant. In cost matrix 5.4, all applicants predicted to Default will automatically be rejected, while the applicants who are predicted as Non-Default will be granted credit. For instance, in figure (b), deployment of the model could result in a potential loss of income of ≈73,000 NOK when rejecting Non-Defaulting Applicants (FP). However, this also results in an avoided loss of ≈92,000 NOK by not granting credit to Defaulting Applicants

(TP). Even though the performance according to the MCC score is seemingly unchanged, the model with the goal of predicting Defaults performs marginally better with the added data from Debt Registry as it captures more of the actual Defaults, which is illustrated by a heightened recall when comparing figure (a) with (b).

Scenario	Realized Loss	Avoided Loss	Realized Income	Missed Income
(a) Without DR	128,000	84,000	1,807,000	61,000
(b) With DR	116,000	92,000	1,795,000	73,000

Table 5.4: Cost matrix: Debt Registry (DR)

However, the model with additional data from the Debt Registry, does not result in a higher profit (equation 5.4 and 5.5). Due to the added loss of income from rejecting more Non-Defaulting Applicants, the increase in avoided loss does not result a change in the profit[36].

$$\textbf{Realized Income} - \textbf{Realized Loss} = \textbf{Profit}$$

$$1,807,000 - 128,000 = 1,679,000 \quad \text{Without DR} \tag{5.4}$$

$$1,795,000 - 116,000 = 1,679,000 \quad \text{With DR} \tag{5.5}$$

$$\Delta = 0 \tag{5.6}$$

As the goal is to predict Defaults, the model ultimately performs better with additional data from the Debt Registry. Presumed that the company does not keep the declined credit idle, but rather grants the credit amount to other future applicants, concerns related to increased loss of potential customers have not been considered. Due to both internal compliance and willingness to take risks, the Company may also be reliant on the information gathered in the Debt Registry. Additionally, information from the Debt Registry might be required to ensure that the Company complies with the regulatory requirements concerning Consumer Debt.

[36] The cost of obtaining the added information from the Debt Registry has not been taken into consideration when calculating the Δ profit.

6 Conclusion

This chapter concludes the thesis and provides reflections made by the authors with respect to potential weaknesses of the thesis.

6.1 Answer to Research Question

The research question raised in this thesis was:

Can a machine learning model create value by predicting default at the time of credit application?

The model correctly predicts 44% of the Defaulting Applicants, which comes at the expense of 4% misclassification of Non-Defaulting Applicants. The avoided future loss does, however, outweigh the potential loss of income compared to the current process. Hence, the machine learning model can create value by predicting default at the time of application.

In extension of the research question, the following subquestions were:

1. *Can the predictions be used to reduce future monetary losses?*
2. *Does data from the Debt Registry increase the predictive performance?*

As the avoided future loss outweights the potential loss of income, the model results in a potential yearly gain of 380,100 NOK. Thus, the model can be used to reduce future monetary losses. Additionally, with data from the Debt Registry, the model correctly predicts 4% more of the Defaulting Applicants without compromising the profit. Consequently, the data from the Debt Registry increases the predictive performance.

6.2 Potential Weaknesses

This subsection provides reflections made by the authors with respect to potential weaknesses of the data and the model.

6.2.1 The Data

When evaluating the model and results, it is important to once again mention that the data solely consists of the applicants that were **granted** credit. Hence, if the model was implemented at an earlier stage in the process, it is unknown how this would affect the performance. For the results to be applicable to real life applications, the model has to be implemented at the last stage in the current application process [37].

Additionally, the data only contains applicants who were granted credit in the period November 2019 - February 2020. As the last application was approved in February 2020, and Norway went into lock down in March 2020, the data solely exists of applicants who were approved right before the global pandemic broke out in Norway. The pandemic may have impacted the applicants behaviour during the 12 months. Hence, the patterns in the data may not be representative for other years as many leisure activities have been limited such as bars, restaurants, and travelling (Finans Norge, 2021b). Additionally, the majority of the Norwegian citizens have been advised to work from home, which could lower the overall travel expenditures. Furthermore, from the 1st-, in 2020, to the 3rd quarter the unemployment rate increase from 2.6% to 5.4%, before it stabilized at 5% in the 1st quarter of 2021 (SSB, 2021). The applicants with secure employment throughout the period might have experienced an increased income to expenditures ratio, and might have prioritized repaying debt (Susanne Solberg Nilssen, 2021). Whereas numerous people who experienced forced temporary leave may have experienced financial difficulties.

Prior to the debt registry, Defaulting Applicants may have been characterized by misleading representation of their debt by underreporting the actual amount. After the implementation of the debt registry, financial institutions are able to confirm the amount of debt registered to an applicant. Hence, one of the reasons for the low value of the Debt Registry might be a change in behavior from the applicants as they may be more willing to report

[37] To ensure that the input data has undergone the same amount of prerequisite filtering as the data on which the model is trained on.

the correct information. On the other hand, as the citizens total debt now can be found in one registry, citizens might also report more accurately due to increased awareness of their financial situation (Karl Wig, 2019).

6.2.2 The Model

Originally, the intention was to keep a minimum of 100 instances of the minority class present in both the training- and holdout data. However, as the data was vastly imbalanced and the model required more data to train on, we settled for only 54 observations in the holdout data. Hence, the model was only validated on a holdout set with of 54 instances of the minority class, which could affect the results when deployed.

Additionally, we acknowledge that the explainability of a machine learning algorithm is limited. However, as long as the model performs better than the current process, it is considered useful. The limited explainability could still be a problem if applicants request an explanation for why they were declined. Nevertheless, when this concern was brought to the Company's attention, it became evident that their decisions are frequently based on tools which provide a low degree of explanation.

Furthermore, this thesis has not taken into consideration the probability of loss compared to the potential amount that may be lost. It should be noted that applicants pose various degrees of risk and that the Company presumably are more concerned about applicants with a high potential loss. Additionally, applicants with high and low credit limits might default based on different reasons that could be overlooked by the model.

Finally, the cost associated to implement and maintain the model has not been considered in this thesis. Neither have the costs and potential reduction of manual processing of applications.

References

Abu-Rmileh, A. (2019, February 8). *The multiple faces of 'feature importance' in xgboost.* Retrieved from https://towardsdatascience.com/entropy-and-information-gain-in-decision-trees-c7db67a3a293

Al-Masri, A. (2019, June 22). *What are overfitting and underfitting in machine learning?* Retrieved from https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690

Badr, W. (2019a, August 6). *3 different ways to tune hyperparameters (interactive python code).* Retrieved from https://towardsdatascience.com/3-different-ways-to-tune-hyperparameters-interactive-python-code-87548d7f2365

Badr, W. (2019b, January 18). *Why feature correlation matters . . . a lot!* Retrieved from https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4

Barne- og familiedepartementet. (2019, July 5). *Gjeldsinformasjonsloven.* Regjeringen.no. Retrieved from https://www.regjeringen.no/no/tema/forbruker/gjeldsinformasjonsloven/id2510537/

Bronshtein, A. (2017, May 17). *Train/test split and cross validation in python.* Retrieved from https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

Brownlee, J. (2019, August 12). *A tour of machine learning algorithms.* Retrieved from https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Brownlee, J. (2020a). *Choose better metrics, balance skewed classes, and apply cost-sensitive learning* (Vol. v1.2). Author.

Brownlee, J. (2020b, February 10). *A gentle introduction to threshold-moving for imbalanced classification.* Retrieved from https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/

Brownlee, J. (2020c, August 15). *Logistic regression for machine learning.* Retrieved from https://machinelearningmastery.com/logistic-regression-for-machine-learning/

Brownlee, J. (2020d, August 26). *Train-test split for evaluating machine learning algorithms.* Retrieved from https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

Chicco, D., Tötsch, N., & Giuseppe, J. (2021, February 04). The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. , *13 (2021).* Retrieved from https://link.springer.com/article/10.1186/s13040-021-00244-z

Finans Norge. (2021a, January 29). *Det haster med å revidere gjeldsordningsloven.* Retrieved from https://www.finansnorge.no/aktuelt/nyheter/2021/01/det-haster-med-a-revidere-gjeldsordningsloven/

Finans Norge. (2021b, February 1). *Markant nedgang i rammekreditter.* Retrieved from https://www.finansnorge.no/aktuelt/nyheter/2021/02/markant-nedgang-i-rammekreditter/

Finanstilsynet. (2019, April 25). *Krav til finansforetakenes utlånspraksis for forbrukslån. (rundskriv 5/2019).* Retrieved from https://www.finanstilsynet.no/contentassets/adff29a42bcc4584acd7883a73e9eef1/krav-til-finansforetakenes-utlanspraksis-for-forbrukslan.pdf

Finanstilsynet. (2020). *Utviklingen i forbruksgjeld 2020.* Retrieved from https://www.finanstilsynet.no/contentassets/fa2c66c4edab430186b90fe1ec743c34/utviklingen-i-forbruksgjeld_2020.pdf

Gjeldsinformasjonsloven. (2017, June 16). *Lov om gjeldsinformasjon ved kredittvurdering av privatpersoner, lov-2017-06-16-47.* Lovdata.no. Retrieved from https://lovdata.no/dokument/NL/lov/2017-06-16-47#KAPITTEL_5

Haugan, I. (2020, November 20). *Norske husholdninger ligger i verdenstoppen i privat gjeld.* Retrieved from https://forskning.no/ntnu-partner-penger/norske-husholdninger-ligger-i-verdenstoppen-i-privat-gjeld/1770716

Kantega. (n.d.). *Færre i gjeldsklisteret med maskinlæring.* Retrieved from https://www.kantega.no/prosjekter/faerre-i-gjeldsklisteret-med-maskinlaering

Karl Wig. (2019, August 12). *Banker etter ny innstramming: Kredittkort skrotes og forbrukslån avslås.* Retrieved from https://e24.no/privatoekonomi/i/dOb02O/banker-etter-ny-innstramming-kredittkort-skrotes-og-forbrukslaan-avslaas

Keboola. (2020, August 24). *The ultimate guide to logistic regression for machine learning.* Retrieved from https://www.keboola.com/blog/logistic-regression-machine-learning

Kho, J. (2018, October 19). *Why random forest is my favorite machine learning model.* Retrieved from https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706

Lindorff. (n.d.). *Alt du trenger å vite om betalingsanmerkninger.* Retrieved from https://www.lindorff.no/kundeservice/tips-til-bedre-okonomi/artikler/alt-du-trenger-a-vite-om-betalingsanmerkninger/

Lutes, J. (2019, November 15). *Entropy and information gain in decision trees.* Retrieved from https://towardsdatascience.com/entropy-and-information-gain-in-decision-trees-c7db67a3a293

Mumtaz, A. (2020, July 30). *How to effectively predict imbalanced classes in python.* Retrieved from https://towardsdatascience.com/how-to-effectively-predict-imbalanced-classes-in-python-e8cd3b5720c4

Narkhede, S. (2018, June 26). *Understanding auc - roc curve.* Retrieved from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Nikulski, J. (2020, March 16). *The ultimate guide to adaboost, random forests and xgboost.* Retrieved from https://towardsdatascience.com/the-ultimate-guide-to-adaboost-random-forests-and-xgboost-7f9327061c4f

Norsk Gjeldsinformajson. (2021, May 31). *Overraskende nedgang i forbruksgjeld i desember.* Retrieved from https://www.norskgjeld.no/statistikk

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011b). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking* (1st ed.). O'Reilly.

Regjeringen. (2017, April 4). *Spørsmål og svar om endring av regler for markedsføring av kreditt.* Retrieved from https://www.regjeringen.no/no/tema/okonomi-og-budsjett/finansmarkedene/sporsmal-og-svar-om-endring-av-regler-for-markedsforing-av-kreditt/id2548055/

Regjeringen. (2019, February 12). *Regjeringen innfører nye krav til banker som tilbyr forbrukslån.* Retrieved from https://www.regjeringen.no/no/aktuelt/regjeringen-innforer-nye-krav-til-banker-som-tilbyr-forbrukslan/id2628803/

scikit-learn 0.24.2. (2021). *1.13. feature selection.* Retrieved from https://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection

Senapati, D. (2018, August 29). *Grid search vs random search.* Retrieved from https://medium.com/@senapati.dipak97/grid-search-vs-random-search-d34c92946318

Sparebank-1. (n.d.). *Om sparebank 1-alliansen.* Retrieved from https://www.sparebank1.no/nb/bank/om-oss/om-banken/om-sparebank-1-alliansen.html

SSB. (2021, May 05). *Arbeidskraftundersøkelsen.* Retrieved from https://www.ssb.no/arbeid-og-lonn/statistikker/aku/kvartal

Sucarrat, G. (2017). *Metode og Økonometri en moderne innføring* (2nd ed.). Fagbokforlaget.

Sullivan, L. (n.d). *Hypothesis testing - analysis of variance (anova).* Boston University School of Public Health. Retrieved from https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_hypothesistesting-anova/bs704_hypothesistesting-anova_print.html

Susanne Solberg Nilssen. (2021, January 05). *Overraskende nedgang i forbruksgjeld i desember.* Retrieved from https://finansavisen.no/nyheter/personlig-okonomi/2021/01/05/7603545/overraskende-nedgang-i-forbruksgjeld-i-desember

Thon, B. E. (2014, November 18). *Nei til gjeldsregister!. personvernbloggen.* Retrieved from https://www.personvernbloggen.no/2014/11/18/nei-til-gjeldsregister/

XGBoost Python Package. (n.d.). *Python api reference.* Retrieved from https://xgboost.readthedocs.io/en/latest/python/python_api.html#

Yiu, T. (2019, July 20). *The curse of dimensionality.* Retrieved from https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e

Appendix

A1 Figures

Figure A1.1: BI logo

A2 Tables

Variables	Data type
SK_APPLICATION_ID	int64
BK_ACCOUNT_ID	int64
BK_APPLICATION_CD	object
NCProduct	object
PeriodId	int64
TheAge	int64
DEBIT_CARD_IND	int64
ApplicationHour	int64
ApplicationWeekDay	int64
ApplicationMonth	int64
ApplicationWeek	int64
EMPLOYMENT_TYPE_NAME	object
HABITATION_TYPE_NAME	object
MARITAL_STATUS_NAME	object
GROSS_INCOME_AMT	int64
IncomeCat	object
WEALTH_AMT	int64
wealthcat	object
DEBT_RATIO_AMT	float64
MORTGAGES_AMT	int64
CONSUMER_LOAN_AMT	int64
VEHICLE_LOAN_AMT	int64
STUDENT_LOAN_AMT	int64
ConsumerLoanCat	object
ConsumerDebtRatio	float64
VEHICLE_LOAN_AMT.1	int64
AllLoansAmt	int64
difftax	int64
avgdifftax	float64
missingtaxinc	int64
DebtRegisterNum	int64
DebtRegisterCreditFacilityNum	int64
DebtRegisterRepaymentLoanNum	int64
DebtRegisterCreditLimit	int64
DebtRegisterIELA	int64
DebtRegisterNonIELA	int64
DebtRegisterOrigBalance	int64
DebtRegisterRepaymentLoanBalance	int64

Table A2.1: Original variables in application data part 1

Variables	Data type
DEBT_NEGOTIATION_IND	int64
DEFAULTED_IND	int64
MORTGAGE_IND	int64
LOGINS_NUM	int64
TRANSACTIONS_NUM	int64
TOTAL_DEBT_AMT	float64
TOTAL_DEPOSIT_AMT	float64
TOTAL_INCOMING_AMT	float64
NUM_OF_ACTIVE_CREDIT_CARDS_CNT	int64
NoOfChildren	int64
dayssincemove	int64
NUMBER_OF_ADDRESS_CHANGES_CNT	int64
PerCnt	int64
Mortgage_exp	int64
Sum_expenses	int64
Stress_Sum_expenses	int64
FLI_AMT	float64
SFLI_AMT	float64
mediumFliInd	int64
monthlyincratio	float64
HowHouse	int64
GRANTED_CREDIT_LIMIT_AMT	int64
APPLIED_CREDIT_LIMIT_AMT	int64
CollectionFirst12Ind	int64
BalanceSentAmt	float64
CumProfit	float64

Table A2.2: Original variables in application data part 2

Model parameters	Grid
Logistic Regression	
Penalty	L2, Elasticnet
Class weight	Balanced, {**0:** 1, **1:** 1, 4, 10, 30, 50}
Random Forest	
Min samples leaf	Random integer ∈ [10,30]
Max depth	Random integer ∈ [3,7]
N estimators	Random integer ∈ [40,150]
Class weight	Balanced, {**0:** 1, **1:** 1, 4, 10, 30, 50}
XGBoost	
Booster	DbTree, Dart
Colsample by tree	Random float ∈ $[0.01, 0.7]$
Eta	Random float ∈ $[0.2, 1]$
Evaluation metric	Logloss, Error
Max depth	Random integer ∈ $[3, 7]$
Min sample leaf	Random integer ∈ $[10, 30]$
Weight minority class	Random Integer ∈ $[1, 50]$

Table A2.3: Grid of all hyperparameters

Parameter	Value
Booster	Dart
Colsample by tree	0.206
Eta	0.212
Evaluation metric	Logloss
Max depth	3
Min sample leaf	23
Weight minority class	36

Table A2.4: Optimal combination of hyperparameters XGBoost