Machine learning vs logistic regression in credit scoring: A trade-off between accuracy and interpretability?

Arne Hovdenakk

Master thesis

The thesis is submitted to fulfil the requirements for the degree:

Master's in Economics

University of Bergen, Department of Economics

June 2021

UNIVERSITETET I BERGEN

Preface

I would like to thank my thesis supervisor Associate Professor Thomas de Haan from Department of Economics. Thank you for good guidance throughout the semester.

I would also like to thank the bank who provided the dataset, and the employees I have had contact with. I could not write this thesis without your good help, and access to the data. Thank you for our good cooperation.

With this thesis five fine years at Department of Economics has come to an end. I need to thank Even and Vegard for good friendship from day one. These years would not been the same without you.

Abstract

In this thesis, I compare logistic regression to the machine learning models k-nearest neighbor, decision trees, random forest, and gradient booster by creating different credit models. By using data from an anonymous Norwegian bank for consumer loan borrowers, I compare the models when continuous variables are split into intervals by using weight of evidence, and when they are kept in their raw form.

By using Area under Receiver Operating Characteristic (AUROC) and Brier score as performance measures, I find that logistic regression and gradient booster are the most accurate models for this dataset, and logistic regression is recommended because of its interpretability.

Contents

List of tables

List of figures

1. Introduction

The purpose of this thesis is to investigate how machine learning models performs compared to a traditional credit model built by using logistic regression. By using two datasets provided by an anonymous Norwegian bank five models are created to predict the behavior of existing borrowers who have consumer loan in this bank. The machine learning models used are k-nearest neighbor, decision tree, random forest, and gradient booster classifier. All the models are trained two times by using the dataset where first; the continuous variables are in their raw form and second; by splitting continuous variables into logical intervals by using weight of evidence where each interval is transformed to dummy variables. There are several other machine learning models who also could be investigated, but these models are chosen because of their interpretability and does not require much feature engineering.

1.1 Introduction to credit scoring

The banking sector has an important role in every modern economy. Every day, millions of banks all over the world must make important decision when they decide to offer credit to both consumers and businesses. If a bank is being too strict and refuse borrowers who would repay their obligations it may miss potential revenue, and the economy will suffer from lost potential given that they are not granted credit any other place. However, if a bank accepts too many applicants, and a high number of borrower's default, it can cause major economic consequences both for the bank and the economy.

To make the decision of granting credit easier, the standard in the credit industry is to use credit models to calculate a credit score when they take these decisions. A credit model is a statistical model designed to predict the behavior of the applicants. Based on the characteristics of the applicant, the finance institution calculates a credit score or a probability of default. The credit models are built using internal data and external data on previous customers. There are many different statistical techniques which can be used to create a credit model, but the most used model in the industry is logistic regression (Crook, Edelman and Thomas, 2007). By using logistic regression, it is easy to transform the coefficients into a credit scorecard (Anderson, 2007). A credit scorecard could be created and shaped in many ways, but a simple scorecard consists of different variables which is separated into groups or intervals where each group gives a sum of points (Siddiqi, 2017). When predicting the behavior of an individual, the characteristics of the individual are plotted into the scorecard which gives a total score. There are several arguments why the scorecard format is so popular

(Siddiqi, 2017). It is easy to understand, interpret and use for both the lender and the customers.

Table 1.1: Example of a scorecard

Characteristic	Bin	Points	Borrower X
Income	0-15.000.-	20	
	15.000-30.000.-	27	x
	>30.000.-	40	
Credit utilization	< 60 %	34	
	60 % - 90 %	28	
	> 90 %	15	x
Length of customer relationship	< 3 months	18	x
	3-12 months	35	
	>12 months	44	
Has previous reminders	No	34	x
	Yes	8	
Total			94

The figure above shows an example of a made-up scorecard with four characteristics. Based on borrower X's characteristics he has 94 points. Then, the bank could transfer the credit score into a probability of default.

It is normal to separate credit modeling in to two types: application score and behavior score. Application score is the score calculated to decide whether to grant credit or not. When deciding whether to grant credit to an applicant, the bank may only accept applicants with a credit score higher than a specific threshold. Behavioral score is calculated after the credit has been granted and should be recalculated regularly. Behavioral score is used to have control over the risk of the portfolio and provisions for bad debts. It can also be used as a tool for the bank if the borrowers apply for new loans. After the loan is granted, the bank has information on the behavior of the borrower. An example of behavior variables are data on reminders, how much of the credit is yet to be paid back and how long time they have had the product. When creating a behavioral scoring model, these variables are also used. In this thesis I will create models of the latter type for existing borrowers who has a consumer loan at the bank.

If the bank has a good model to predict whether a borrower default or not, the risk management of the bank becomes easier and more efficient. If the bank early discovers "bad" borrowers (borrowers who later turns out to default their loan) in their portfolio, they could adjust their provisions for bad debts to a more precis level. This would make it easier for the

bank to plan how much new loans they could offer in the future and allocate capital more efficient. It also makes it easier for the bank to take actions against the predicted bad borrowers to reduce the risk of default.

The purpose of this thesis is to investigate whether machine learning models could improve the accuracy of the credit models. Machine learning is created to find patterns in data which human beings are not able to. However, some machine learning models are considered a "black box", because it is not always easy to explain the results. Therefore, it is not always an easy task to understand what truly separates the good borrowers from the bad when using some models. A deeper discussion of this is done in the part of machine learning.

In this thesis, I will create several credit models using different techniques. A traditional credit model is created where logistic regression is used where the continuous variables are transformed into intervals based on their weight of evidence. Then I will compare logistic regression to the other machine learning models by using Brier score, and the area under the Receiver Operating Characteristic curve as performance measures. I will also compare the models when the continuous data is kept in their original forms except the missing values which are treated.

1.2 Outline of the thesis

The thesis is organized as follows: Chapter two is an introduction to machine learning in general. Chapter three is a review of related literature on machine learning in credit scoring. Chapter four is a description of the methodology used when creating the models and includes a description of the models used. Chapter five describes how the variables are engineered and variable selection. Chapter six presents the results and a model to illustrate the economics in credit scoring. Chapter seven summarize the findings and gives a recommendation to the bank.

2. Machine learning

2.1 What is machine learning?

Machine learning can be defined as algorithms who by being fed input data is trained to perform regression or classification, some sort of grouping and clustering of data (Athey, 2018). The algorithms are "trained" by being fed the input data. There are many different types of models developed for these tasks. Within machine learning, we normally have two main types of machine learning: supervised and unsupervised learning. When using unsupervised learning, the target variable Y is unknown. Among the most famous unsupervised algorithms, we have clustering algorithms like k-means clustering. An area where unsupervised learning is much used is within face recognition (Müller and Guido, 2017).

Supervised learning is more like traditional econometrics in the way that the target variable is known. While econometrics often focus on finding the causal inference, after a "treatment" or a change in policy supervised machine learning is mainly used for prediction (Athey, 2018). Machine learning has a more "practical" approach to problems, while econometrics have a more academic approach (Iskhakov, Rust and Schjerning, 2020). In econometrics, the goal is often to do estimation based on econometric theory.

We distinguish from individual classifiers/regressors and ensemble learners. Ensemble learners is two or more algorithms doing the estimation. Among the ensemble learners, we separate between homogenous ensembles (the same algorithms estimating several times) and heterogeneous ensembles (where different types of algorithms are used to estimate). Within the different machine learning models, there are parametric and non-parametric models. Among the parametric models we find LASSO and elastic net. Examples of non-parametric models are random forest and k-nearest neighbor (Mullainathan and Spiess, 2017). While econometrics models focus on the parameters, for example the coefficient β in a regression, and their impact to the target variable, the machine learning models are not created for a purpose like this (Mullainathan and Spiess, 2017). Machine learnings models are built to discover complex relationships and patterns that humans could not discover.

The challenge of credit scoring is mainly to predict the behavior of the borrower. It is a binary problem where it is normal to separate between good (no default) and bad borrowers (default). As discussed above, supervised machine learning models could be used for this purpose. A

more accurate model could be very beneficial for a credit provider. Therefore, banks could have great benefits applying machine learning techniques if they increase the credit model's ability to predict the behavior of the borrower.

2.2 Overfitting

The risk of overfitting is a higher concern in machine learning than in traditional econometric literature (Athey and Imbens 2019). Machine learning algorithms uses the input data to learn and train the algorithms. If the performance is significantly lower when testing the trained algorithm on (similar) new data (held-out-sample), the algorithm has over-fitted the training data. This is not desirable, because the goal is to create robust models who can predict new data. The problem of overfitting is a potential problem in machine learning because the machine learning models wants to create flexible complex models to best predict the target variable.

One way of controlling for this potential problem is to use cross-validation (Athey and Imbens, 2019). Before training the models, some part of the data is held outside the training process. This out-of-sample data is then used to check how the machine learning models perform after the models has been trained. In addition, some models also reduce the potential overfitting problem by averaging several models. A detailed explanation of this is done when the models used in this thesis is discussed in chapter four.

3. Literature review

There is a large body of literature within the area of comparing machine learning models to the benchmark model logistic regression in credit scoring. According to (Crook, Edelman and Thomas, 2007), the modern history of credit scoring starts with David Durand's article from 1941, where he used statistical techniques to separate good and bad loans to firms. Later, especially the second half of the 20th century linear probability modeling and discriminatory analysis where the most used statistical techniques in credit scoring (Anderson, 2007). In modern times, logistic regression is the most used model in credit scoring (Crook, Edelman and Thomas, 2007).

Other, more complex techniques have also been investigated. Baesens et al. (2003) looks at support vector machines, neural network, decision trees, k-nearest neighbor, linear programming, and Bayesian network classifier. Random forest is also a model who has been applied (Kruppa et al., 2013), but logistic regression is considered a standard in the industry. When creating a credit model, it is often a trade-off between having an accurate model and a model which is possible to interpret (Florez-Lopez and Ramon-Jeronimo, 2015). Logistic regression has a benefit of both being accurate and possible to interpret.

The different papers use numerous different models to compare the accuracy of the models created for the binary classification problem of credit scoring. When highlighting the conclusions from the literature, I will highlight the findings for the models used in this thesis, which is k-nearest neighbor, decision trees, random forest, and gradient booster. Lessmann et al. (2015) uses individual classifiers, homogenous and heterogeneous ensembles when comparing models. They use eight different datasets to compare in total 41 classifiers. Lessmann et al. (2015) finds that random forest performs better than logistic regression, stochastic gradient boosting model and k-nearest neighbor. A stochastic gradient boosting model differ from gradient booster by using a smaller subsample to create the trees. As a conclusion Lessmann et al. (2015) recommends that random forest should be used as a benchmark to compare new classifiers.

Kruppa et al. (2013) compares logistic regression to random forest and k-nearest neighbor. They find that random forest outperforms the two other models significant. Random forest has an AUROC (Area under the ROC curve) of 0,959 and Brier score of 0,071, compared to logistic regression with an AUROC of 0,748 and Brier score of 0,11 and k-NN (k-nearest neighbor) with an AUROC of 0,685 and Brier score of 0,116.

Finlay (2011) finds that Logistic regression performs better than decision trees (Classification and regression trees) and k-NN.

Liu, Fan, and Xia (2021) uses six relatively small datasets to assess the performance of the different models. They find that gradient booster decision tree (AUROC: 0,9311 and Brier score: 0,1078) performs just as good as both random forest (AUROC 0,9321 and Brier score 0,1038) and logistic regression (AUROC 0,9217 and Brier score 0,1050). Decision trees (AUROC 0,8194 and Brier score 0.1903) has the lowest accuracy among the selected models from their study.

Brown and Mues (2012) analyzed the effect of different grade of imbalanced data on the performance of the different models. By over- and under-sampling of the good and bad cases, they created six different bad rates (default rates), from 30 % to 1 %. The tree based ensembled model's gradient boosting and random forest performed good for extreme levels of imbalances. Logistic regression is not far behind these models. K-NN and decision trees performance fell when the level of imbalance where more extreme.

Mostly in credit scoring, the focus is how to gain the highest accuracy. In the literature in general, there is less or no focus on interpretability (Florez-Lopez and Ramon-Jeronimo, 2015). Florez-Lopez and Ramon-Jeronimo (2015) addresses this issue and concludes that of the 24 studies they investigated, only one third of the studies included some interpretability measure like for example feature importance. In their paper, they propose a correlated adjusted decision forest (CADF) where the goal is to gain both high accuracy by applying ensemble strategies and interpretability. They conclude that the new model performs about just as good as random forest and gradient booster, and much better than k-NN and logistic regression. Florez-Lopez and Ramon-Jeronimo (2015) concludes that the new CADF is not outperformed by the "black-box" models' random forest and gradient boosting regression and CADF is also possible to interpret.

Summarized from the findings from the related literature, random forest and gradient booster performs best. However, logistic regression is in many cases not far behind. The simpler models k-nearest neighbor and decision tree has in general the lowest performance.

4. Methodology

4.1 Consumer loan

The aim of this thesis is to create a good economic forecasting model to predict the behavior of consumer loan borrowers. Consumer loan is an unsecured loan mainly used for consumption, but the usage areas for this type of loan varies. A report from Poppe (2017) finds that the most frequent purposes for this credit type in Norway turns out to be consumer goods, unexpected expenses, payment on existing debt and travels. In other words, there are individuals in different economic situations among the consumer loan borrowers. According to Finanstilsynet (2020a), the average default rate for Norwegian consumer loans was 13,4 % by the end of second quarter of 2020. This rate has increased by almost 2,4 % points since the end of 2019. The economic crisis following the global pandemic could be a significant reason to the increased default rate.

A consumer loan is typically a short-term loan, but some cases it is possible to have a maturity over several years. The interest rate is higher than secured credits like mortgages, but the interest rate varies from different lenders and maturity time.

4.2 Definition of default

When creating a model where the goal is to predict an outcome, it is important to have a clear definition of the actual target variable. In credit scoring the goal is to predict whether a borrower is defaulting or not. The financial regulators have strict requirements of what they define as default when creating credit models. Finanstilsynet (2020b) defines default as:

- If a borrower is at over 90 days late past due the borrower is considered a default case.
- If the bank consider that the borrower is not likely to repay his dept, the borrower is considered a default case.

In the dataset used in this thesis, the performance window are 12 months after a snapshot of the characteristics of the borrower. If a borrower is more than 90 days past due within this performance window, it is considered a default. In this thesis non-default and default will also be referred to as good and bad outcomes.

4.3 Software

For all the work with data preparation, training and testing the models the programing language Python is used. Python is a free programming language and open for everyone. By

using the Python software “sklearn”, it is relatively easy to use the different machine learning models, also for non-machine learning experts.

4.4 Models

This section explains the models used to create the credit models in this thesis. A discussion of advantages and disadvantages is also included. Logistic regression is included because this is a popular model within credit scoring. K-nearest neighbor and decision trees are simple machine learning models which is easy to explain. Gradient booster and random forest are more powerful algorithms who also are popular models. These are included because these are more complex models who does not require much data preparation.

4.4.1 Logistic regression

In the banking industry, logistic regression is the most used method when creating a credit model (Crook, Edelman and Thomas, 2007). When using logistic regression, it is possible to estimate the relationship between the independent variables X, and a dependent variable Y when Y is a binary variable. In credit scoring, the aim is to forecast whether a borrower is defaulting or not. Therefore, logistic regression is a useful model for this task.

The form of logistic regression used is expressed by the probability of our dependent variable Y is one given the features, X.

$$\Pr(y=1|X_1,\ldots,X_k) = \frac{e^{\beta_0+\beta_1X_1+\ldots+\beta_kX_k}}{1+e^{\beta_0+\beta_1X_1+\ldots+\beta_kX_k}} = p$$

Where p is the probability for the outcome y equals one, and thus one minus p is the probability for y to be zero. When using logistic regression in econometrics, the logit transformation is applied (Bolton, 2009), which yields:

$$\frac{p}{1-p} = e^{\beta_0+\beta_1X_1+\cdots+\beta_kX_k}$$

$$\text{Ln}\left(\frac{p}{1-p}\right) = \beta_0+\beta_1X_1+\ldots+\beta_kX_k$$

The dependent variable is log the odds of an event accruing (the probability of the event divided by the probability of the non-event). Using the logit transformation has many benefits. It has many of the characteristics of linear regression (Bolton, 2009). The dependent variable may be continuous and linear in its parameters. The coefficients are estimated by using maximum likelihood estimation.

Logistic regression has five main assumptions (Anderson, 2007):

- The target variable is categorical.
- Independent error terms.
- The predictors are uncorrelated.
- Linear relationship between log the odds of the target variable being one and the independent variables.
- The variables are relevant.

One of the benefits of using logistic regression is that it has proven to be relatively accurate in credit scoring (Anderson, 2007). It is possible to interpret and explain why it classifies a case, which is a benefit to in credit scoring. It is also possible to transform the coefficients to a scorecard format.

A downside of logistic regression is that it may not find complex patterns and relationship like other machine learning models may do. This is especially a weakness if the data is complex and noisy. Logistic regression also requires data preprocessing. It cannot handle missing data, and to capture the non-linear effects of some variables, they must be transformed.

4.4.2 K-nearest neighbor

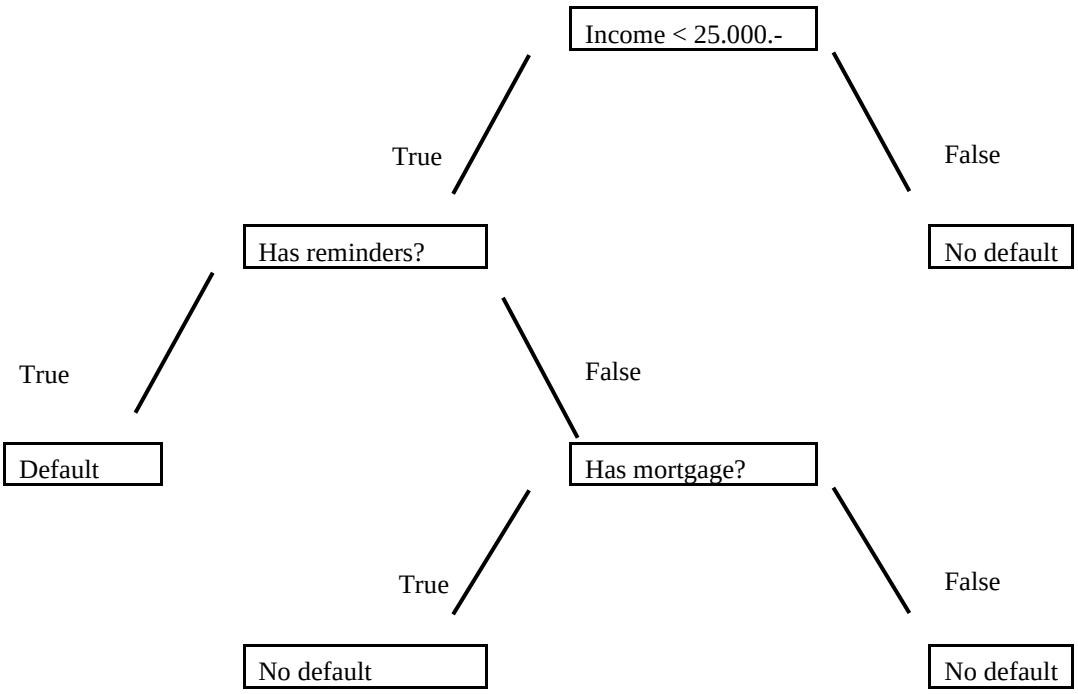
K-nearest neighbor (k-NN) is a non-parametric machine learning model who classifies new data based on the nearest datapoints in the training dataset (Müller and Guido, 2017). The algorithm finds the k nearest datapoints based on the features. The k-NN algorithm predicts the class of the new data based on which class the nearest neighbors belong to. The k in k-NN is the number of neighbors the new datapoint is compared to (Müller and Guido, 2017). If for example k is seven, the algorithm looks at the seven closest neighbors based on the features. The new data is classified by the highest votes among the neighbors. For instance, if six of the neighbors are in class “good”, and one in class “bad”, the new data is classified as “good”.

K-NN is a simple and easy machine learning model to explain to non-experts. The fact that k-NN is one of the simplest models could also be a weakness. It may not find the complex patterns that other, more complex model finds. Another downside is that k-NN performs better when data is preprocessed (Müller and Guido, 2017). When having large datasets, the training process could be time consuming. If the training data is large, the algorithm uses long time to search through all the datapoints in the training data. This is one of the reasons why k-NN is not widely used in credit scoring (Bolton, 2009)

4.4.3 Decision trees

Decision trees can also be used for classification. A decision tree is a machine learning algorithm where the goal is to classify the target variable correctly by asking a series of questions using the training data (Müller and Guido, 2017). The illustration below shows an example of a decision tree.

Figure 4.4.3: Example of a decision tree



The way the tree is built starts with the algorithm going through the different features, searching for the best test, which in machine learning is questions who best separates data (Müller and Guido, 2017). An example of a test could be splitting the dataset by age > 30 years. The algorithm chooses the best test (question), which is the top node of the tree. The algorithm continues to separate data by asking additional questions until each leaf consist of only one class. If the tree does not have any restriction of how deep the tree can grow. If there are restrictions of how many questions the algorithm can use to create the tree, some leaves may not be pure (may consist of both good and bad outcomes). When predicting the behavior of a borrower using this algorithm, the predicted behavior (good or bad) is decided by which class has the majority in the leaf this borrower ends up in.

Decision trees is easy to understand and interpret the results. It is possible to have control over which variables separates predicted good from predicted bad. Further, another benefit is that the algorithm is fast when training. This algorithm does not require any data preprocessing to perform.

One disadvantage is that decision tree algorithm tends to overfit the training by building trees which are deep and complex (Müller and Guido, 2017). To prevent this overfitting, a solution could be to set a maximum depth for the tree. This would decrease the accuracy of the training data but increase the accuracy when using the test data. If the data structure is complex, the decision tree must create deep trees to classify the target variable. If this is the case, a very deep and complex decision tree could overfit the training data. Another disadvantage of using decision trees is that the tree is exposed to change drastically if there is a small change in the training data (Hastie, Tibshirani and Friedman, 2009). A small change could lead to drastically change in the splits, and thus decision trees have a high variance.

4.4.4 Random Forest

As mentioned earlier, building a single tree would often lead to overfitting of data. There are two tree-based ensemble machine learning models which could be used to deal with this problem.

Random forest was introduced by Leo Breiman (2001). The random forest algorithm builds a n number of decision trees, where every tree is unique (Breiman, 2001). After several trees are created, all trees give a predicted probability for each class. The average probability for both classes is calculated, and the final prediction of the random forest model is the classification which has the highest average. For example, if we have a borrower *i*, and the output of the forest is an average of 25 % for class good and 75 % for class bad, borrower *i* are classified as class bad.

The algorithm takes a bootstrap sample of the training data (Müller and Guido, 2017) when building a tree. This method is also called “bagging”. This means that the algorithm chooses a n number of data points with replacements. Therefore, some trees are built without somewhat of the training data and could include duplicates of data. A tree is built on this dataset, but the building process is some different from the normal decision tree. In the random forest model, the different trees choses a given number of features, and search for the best test within the chosen features (Müller and Guido, 2017). Therefore, would each tree in the forest be built on

different dataset, and with different sample of features. After all trees are built, the forest consists of independent trees.

By using the bagging technique, the problem of high variance with decision trees is reduced (Hastie, Tibshirani and Friedman, 2009). The model is more robust to changes in the training data. Also, because the model is based on many different random built trees with different variables, the problem of overfitting is significantly reduced (Müller and Guido, 2017). As decision trees, random forest does not require data preprocessing and could handle outliers.

Random forest creates many complex trees. Thus, it could capture complex structures, like non-linear relationships in the data, and still not overfit the training data. But this implies that it is hard to interpret and understand why random forest classify borrowers as good or bad. This a negative factor in credit scoring because it is important to know what the source of risky borrowers is.

The scikit software for Python has a function which ranks the variables by importance. In this way it is possible to rank the variables by how important they are in the building of the trees (Hastie, Tibshirani and Friedman, 2009). This could provide some insight regarding which variables are most important to separate borrowers in the most efficient way and could also be used for further research.

4.4.5 Gradient boosting classifier

Gradient boosting trees does not create as complex trees like the random forest. The idea of gradient boosting trees is that many weak learners (many trees) would combined be a good predictor (Müller and Guido, 2017). The algorithm creates several trees with few questions (few nodes) where the next tree tries to make up for the mistakes made by the previous tree. The depth of the tree is normally not deeper than five (Hastie, Tibshirani and Friedman, 2009). Each tree could turn out to be a good predictor for some of the data, and less good for others. By combining several trees, the performance would improve (Müller and Guido, 2017).

The gradient booster classifier has mainly the same benefits as random forest. It can discover complex data with non-linear relationships and does not require data preprocessing (Müller and Guido, 2017). Because many different trees are created, the probability of overfitting the training data is also reduced.

Gradient booster classifier is a very powerful model (Müller and Guido, 2017). The algorithm needs tuning of the parameters and could be slow when training. This is one of the largest

weaknesses for gradient booster trees. In the same way as random forest, it is not very easy to interpret why this model classify a borrower to be good or bad. Although gradient booster does not have as complex structure as random forest, it is still a complex model. In the same way as for random forest the scikit software for Python has a feature importance function who could be used to increase interpretability. When calculating the variable importance for this model, one could observe that some variables is ignored.

4.5 Hyperparametric tuning

When running the machine learning models there is several hyperparameters who affects how the models search through the input data to find patterns. For the k-nearest neighbor classifier the most important parameter is the number of neighbors (k) used to decide which class new datapoints are predicted to be.

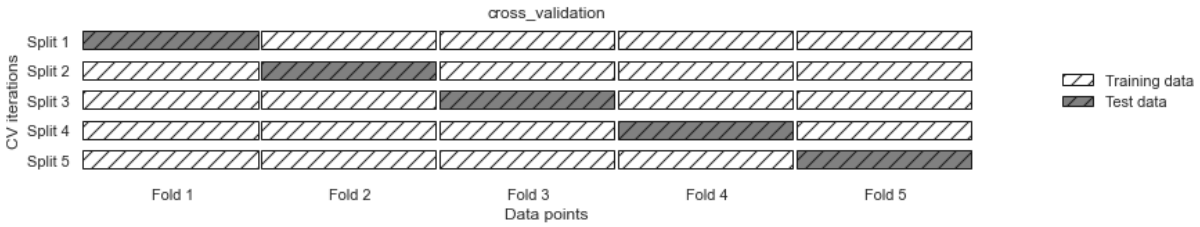
For the decision tree the most important parameter is the depth of the tree. This parameter limits how many questions could be asked to categorize the input variable before the model predicts the outcome variable. A maximum depth of four for the tree of would give the algorithm only four question to classify the input data. If this variable is not regulated, the standard variable is "none". In a situation where this variable is not regulated, the decision tree would overfit the training data.

The random forest algorithm has many parameters, but the most important are maximum depth, and number of trees. Number of trees is the parameter which decides how many trees the random forest algorithm would create. For the gradient boosting classifier, the parameters learning rate and number of estimators is fine-tuned. Number of estimators is the same as for random forest, number of trees in the model. The idea of the gradient booster algorithm is that the trees created tries to correct the errors the previous tree made. The learning rate controls in what grade the new tree created can do that. If the learning rate is low, more trees are needed to create a more robust model (Müller and Guido, 2017).

To find the hyperparameters who creates the most accurate model, a grid search using k-fold cross validation is performed. When using k-fold cross validation, the training data is split into *k* almost equal parts, where one part is the test set, and the rest is used to train the model (Müller and Guido, 2017). This is done *k* times, and the next time the model is trained, another part is used as test data. An important note is that in this process, only the training data is used. The original test data is still being held out, to be used for the final performance evaluation.

The illustration below from Müller and Guido (2017), shows an example of a cross validation with five splits. The model is trained five times.

Figure 4.5: Cross validation (Müller and Guido, 2017)



Using the sklearn package "GridSearchCV", the models use cross validation to find the best combination of hyperparameters who yields the best performance. By using this method, the risk of overfitting is reduced (Müller and Guido, 2017).

There are more parameters who could be optimized then mentioned above, and the possible values for the parameters could be used in the grid search could be wider, but this requires a lot of time and high computer power. Therefore, in this thesis the grid search limits to the parameters and values in the table below.

Table 4.5: Hyperparameters used in the grid search

Model	**Parameter**	**Grid search values**
K-NN	n_neighbors	5, 9, 15
Decision tree	max_depth	3, 5, 9
Random forest	n_estimators	25, 50, 100, 200
	max_depth	3, 5, 15, 20
Gradient Boosting Classifier	learning_rate	0.05, 0.1, 0.2
	n_estimators	100, 200, 300

4.6 Data preprocessing

4.6.1 Description of data

The data used in this thesis is provided from a Norwegian bank. Because of the confidentiality agreement with the bank, the name of the bank will not be revealed. The bank has provided two datasets to be used in this thesis. The first dataset contains information on every borrower who had a consumer loan for a given time. The datapoints used in this thesis are a snapshot of the borrower's current financial situation six months after the consumer loan is approved. For some borrowers, those who had the consumer loan for a less time than six months, the last date of information is included. If there is only information on a borrower for four months,

then the snapshot of the fourth month is included. Also, to make sure that every borrower has had a 12-month performance window, the datapoints from 2020 are not included in the model. If a borrowers 6th month is January 2020, the observation from December 2019 is used instead.

There are 47 explanatory variables in the first dataset. These includes various types of information. The most important is ID, date, length of customer relationship, information on other credit products, payments, and salary for three months backwards, information on used credit on other unsecured loans, number for reminders, size, and utilization of the consumer loan and three default variables.

The second dataset has information on the daily positive and negative balance for the borrower in the bank. The positive balance is the sum of all accounts, like user account and savings account. This dataset also contains information on the negative balance, which consists of the total credit the borrower has in the bank. In both dataset each borrower has an ID number, which makes it possible to connect the datasets.

After the two datasets are merged, the dataset used in this thesis has 12.240 datapoints, where each datapoint represents a unique ID number. The dataset now has a structure like the table below. The data in the table is made up and is not related to the dataset used in the thesis:

Table 4.6.1: Structure of the data used in the analysis.

ID	Date	Approved Date	Average income 3 months	No reminders	……	Default = 0, No default = 1
1	01.07.2019	01.01.2019	23500	0	……	1
2	01.11.2017	01.05.2017	10500	1	……	0
3	02.08.2018	02.02.2018	5400	1	……	0

4.6.2 Missing values

When creating a credit model, the most time-consuming process is to clean and prepare the dataset. One major part of this process is to treat the missing values. In this thesis, I will apply logistic regression for one of the credit models. When using logistic regression, the dataset must be complete without missing values (Siddiqi, 2017). Siddiqi (2017) highlights four methods of treating missing values.

1. The first method is to remove all datapoints including missing values.
2. The second method is to remove all features where more than 50% off all observations are missing.

3. The third method is to create an own group for missing values when binning the variables.
4. The fourth and last method is to use statistical methods to replace the missing values. An example of this method is replacing the missing value with the average value for the dataset.

When choosing a strategy to handle missing values, it is crucial to uncover the reason for the missing variable. If the variable is missing because the borrower does not report the information, one could assume that this information is in the borrower's disfavor. It is irrational for a borrower to not report a variable who is favorable for the individual. In credit scoring one could therefore argue that a missing value could imply negative information.

The dataset provided by the bank has in general few missing values. For the variables who has a small number of missing values (<50), the missing is replaced by the average for the variable. The main strategy used is to treat the missing values as an own group when splitting the continuous data into groups. By using this strategy, it is possible to investigate the effect of the missing value.

When treating the missing values for the dataset where the continuous variables are not separated into intervals, the missing value must be replaced. Because the continuous variables are kept continuous, a separate bin for missing values cannot be created. For the variables relating to the daily balance, it is a significant number of missing values. It is not an alternative to remove all datapoints because that would lead to a big loss of information. It is not clear why these values are missing, but a possible explanation could be that the time for gathering of information is different for these datapoints. The missing values are replaced by an average value for the dataset.

4.6.3 Splitting data

When creating models where the aim is to forecast a behavior, we need to split the dataset into a training and test dataset. The split is done to ensure that the models are not overfitting the training data and give a realistic measure of the performance for the models. By using the "sklearn" package "train test split", the dataset is randomly split into a training and a test dataset.

After the split is done the default rates for the training and test set are a few per mil different. The total default is confidential, but there is a small difference between the groups. The difference is not big, but a chi-square test is done to make sure that the difference in default

rate between the groups is not statistically significant. The split of the data is random, and if the held-out sample has a relatively large proportion of bad borrowers, the model can be biased and therefore perform poor on the test data and future borrowers.

The chi-square statistics is calculated by:

$$\chi^2 = \sum_{i=1}^{k} \frac{(actual - expected)^2}{expected}$$

Where k is the different groups, which is test and training data. Actuals are the number of defaults in each group. Expected for each group k:

$$\text{Expected} = \text{total group}_k * \frac{total\ defoult}{total\ borrowers}$$

With one degree of freedom and a significance level of 0,05 gives 3,841 as critical value for the Chi-Square Distribution.

The chi square test statistics χ^2 are calculated to be 0,80 so we do not reject the null hypothesis. Therefore, we cannot assume that there is significant difference in the distribution of defaults between train and test datasets.

The train and test data contain respectively 80 % and 20 % of the datapoints. The split could be done different, for example with a smaller training group. However, in this thesis an 80/ 20 split is done. This gives a good balance between having a high number of datapoints to create the model, and a sufficient test group. The train dataset is used to engineer the features, select the features to include in the model, select hyperparameters and train the algorithms. Finally, the test dataset is used to evaluate how accurate the models perform and compare the different models.

4.6.4 Feature engineering

In credit scoring it is normal to not use the features in their raw form (Bolton, 2009). Continues variables are often split into logical groups or truncated to reduce impact of outliers. The standard in the industry is to split characteristics into several groups which has logical intervals, often by using weight of evidence (Anderson, 2007). In this thesis, two different datasets using two different techniques are created:

- First, where all continuous variables are in their raw form, except the missing values.
- Second, where logical intervals for each continuous variables are created based on weight of evidence, and then each interval is transformed to a dummy variable.

Some of the more complex machine learning models could want to split the continuous variables different than what is done manually. To investigate the potential difference in the performance between the manually splitting of the variables and continuous data, both versions are tested. This would also give a fair comparing between the different models.

When splitting a continuous variable, it is first split into around 10 groups. These groups are called bins. An example of this binning could be to split the continuous variable income into 10 different bins. Bin 1: all missing values, Bin 2: 0-10 000.- etc. Then, the bin is transformed into a dummy variable which is used in the logistic regression.

By transforming our categorical and continuous variables into logical intervals it is easier to analyze the data. This process is also known as coarse classification. There are several arguments why one should split continuous features instead of doing regression with continuous variables. Some variables do not have a linear relationship with the target variable. An example of this could be age. Intuitively increased age would lead to lower risk of default. Higher age implies all other equal a more robust financial situation. However, the age effect can be expected to be different between a borrower with an age of 20 relative to one with an age of 30, and between a borrower with age of 40 and 50. If using a continuous variable, one would not capture this relationship. One could use income squared and the natural logarithm of age to try a model this relationship, but this would create more complex model, and thereby harder to explain to the borrower. On the other hand, by separating the continuous variable to different bins creates a discontinuity which some would like to avoid. The argument against binning is understandable because it could lead to loss of information. However, Anderson (2007) argues that this could be the best way of handling with the non-linear relationship between the independent variable and the target variable. Creating dummy variables for each bin also makes it easy to allocate points to each coefficient to be used in a credit scorecard.

The binning process gives more information on the dynamics of the data. When binning the variables into different classes one could see what separates the behavior of the different borrowers. Siddiqi (2017) highlights that binning makes it possible to separate missing values so that one bin only consists of missing values. This is a major benefit because some variables have a high level of missing values in this dataset. By doing this one can identity the impact of the missing values for each variable. Binning also makes it possible to reduce the impact of outliers in the dataset. When deciding the cut offs of the different bins, weight of evidence is used.

4.6.5 Weight of evidence

Weight of evidence (WOE) gives an insight into the proportion of good borrowers relative to the proportion of bad (defaulting) borrowers for each bin. This tells us how large the prediction power of each bin has to the dependent variable. A negative WOE implies that the proportion of bad is larger than the proportion of good borrowers for that specific bin. When doing the binning, the WOE value of each bin decides how the final bins are formed. Bins who have equal WOE is merged to one bin. According to Siddiqi (2017), one should start by dividing a continuous variable into several different bins. It is important that each bin has at least 5 % of the population and consist of minimum one of each case (good/bad).

For each feature x, WOE for bin *i* is then calculated using this formula:

$$WOE_{bin\ i} = \ln\left(\frac{\frac{number\ of\ good\ borrowers\ in\ bin\ i}{total\ good\ borrowers\ in\ feature\ x}}{\frac{number\ of\ bad\ borrowers\ \ in\ bin\ i}{total\ bad\ borrowers\ \ in\ feature\ x}}\right) = \ln\left(\frac{Distribution\ of\ good}{Distribution\ of\ bad}\right)$$

Numeric example:

Let us look at a numeric example for 500 borrowers of how WOE is calculated for the for the variable income. This table is made up and is not related to the actual data used in the thesis. This variable separated in to four bins where one bin is for missing values.

Table 4.6.5: Example of WOE calculation

Bin	Total distribution	Good	Distribution of good	Bad	Distribution of bad	Bad rate	WOE
0-15000.-	30 %	110	22,00 %	50	50,00 %	31,25 %	**-0,821**
15000-30000.-	45 %	210	42,00 %	35	35,00 %	14,29 %	**0,182**
30000.- +	20 %	150	30,00 %	5	5,00 %	3,23 %	**1,792**
missing	5 %	30	6,00 %	10	10,00 %	25,00 %	**-0,511**
Total	**100 %**	**500**	**100,00 %**	**100**	**100,00 %**	**16,67 %**	

Where WOE for bin 0-10.000.- is calculated:

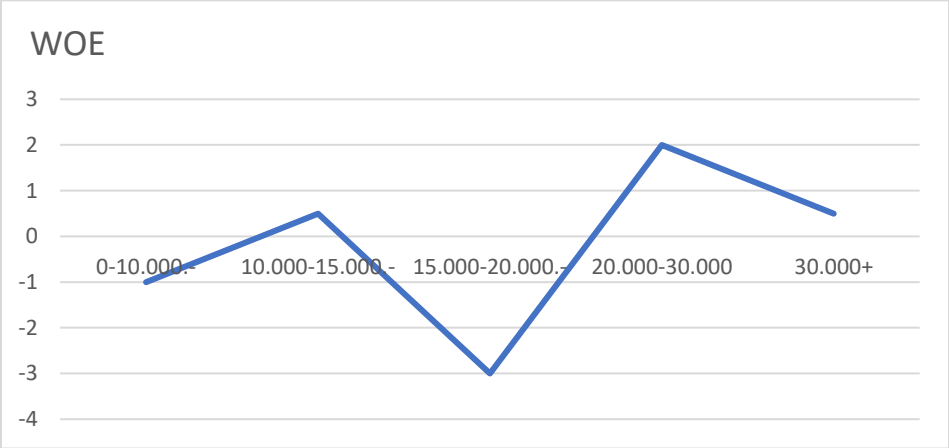
$$WOE_{bin\ 0\text{-}15.000.\text{-}} = \ln\left(\frac{\frac{110}{500}}{\frac{50}{100}}\right) = \ln\left(\frac{22\%}{50\%}\right) = -0{,}821$$

Here, the WOE is increasing with income.

Further on, we calculate the weight of evidence for each bin. The goal is that that the selected features have a logical WOE trend. An example of this could be the feature “income”. Higher income would logically have an increasing weight of evidence curve. It is natural to assume that a higher income would lead to a better economic condition for the borrower and make it easier to fulfill their obligation. If some bins have the same WOE, it implies that the two bins have the same predictive power to the target variable and is therefore combined into one bin. This process is called fine classing of the variables. The WOE curve must be logical, but does not have to linear (Siddiqi, 2017). The goal is to create a good, logical model which is possible to explain in a business sense. If the WOE curve after fine classing is volatile like the curve for a stock price, it is not easy to explain the WOE curve in an intuitive way.

Example of illogical WOE curve

Figure 4.6.5: Example of illogical WOE curve.



If this is the case, it is not possible to explain why the WOE curve is at its form. It makes no sense that the group where income is 15.000-20.000.- has a lower WOE than the lowest income group. If this were the case, this variable would not be included in the model.

4.6.6 Information value

The next step is to calculate the information value for each feature. Information value (IV) is a measure of the total predictive power of a characteristic, x (Siddiqi, 2017). The way to calculate information value is expressed by:

$$IV = \sum_{i=1}^{n}((Distribution\ of\ good - distributon\ of\ bad) * WOE_i)$$

According to Siddiqi (2017), the rule of thumb for interpreting the predictive power of a feature using information value is:

IV < 0,02: unpredictive

0,02–0,1: weak

0,1-0,3: medium

0,3 +: strong

In this thesis, the variables which have an information value of less than 0,02 are excluded from the model.

4.6.7 Correlation

If variables used in logistic regression is highly correlated, the potential problem of multicollinearity could arise, which can reduce the models out-of-sample performance (Anderson, 2007). Therefore, if two variables or more are highly correlated, only one of them is kept in the model.

4.6.8 Variable selection

In this thesis, the Akaike information criteria (AIC) is applied for variable selection by using logistic regression. The Akaike information criteria was formulated by H. Akaike in 1973. The AIC is expressed by this formula (Konishi and Kitagawai, 2008):

$$AIC = -2\,log(maximum\,log-likelihood) + 2\,(number\,of\,parammeters).$$

When using AIC to select models, the AIC is calculated for every proposed model, and then the model who has the lowest AIC is chosen (Hastie, Tibshirani and Friedman, 2009). The Akaike information criteria penalize if the number of variables increase. Thereby, the AIC shows the trade-off between increasing parameters to increase the accuracy of the model and reduce the potential of overfitting when more variables is introduced.

4.7 Evaluation criteria's

To evaluate and validate the models created in this thesis there are several possible criteria which could be used. By using the held-out sample, the test data, the models are evaluated with a new dataset to give a fair evaluation. The area under the Receiver Operating Characteristic curve is one of the most measures used evaluation criteria in credit scoring. In this thesis the area under the ROC curve and the Brier score are used as a performance criterion.

4.7.1 Area under Receiver Operating Characteristic (AUROC)

The goal in credit scoring is to create a model that can distinguish the good from the bad borrowers. A so-called confusion matrix as shown below, tells how accurate a credit model is

at correctly classify the borrowers for a given cutoff. A cutoff is the threshold for which probability default is predicted to be good or bad (non-default and default).

Table 4.7.1: Confusion matrix for a given threshold.

	Predicted		
Actual		Good	Bad
	Good	True positive (TP)	False negative (FN)
	Bad	False positive (FP)	True negative (TN)

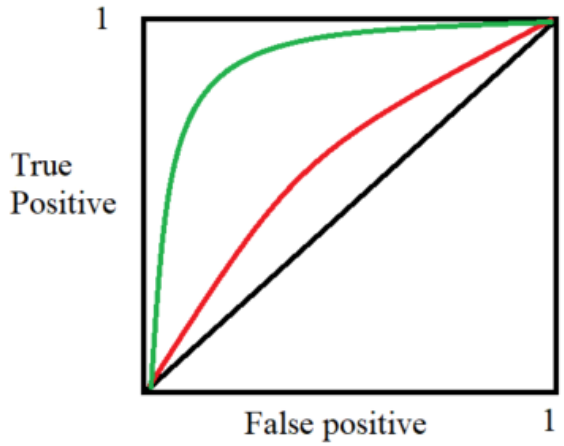
The Receiver Operating Characteristic (ROC) curve is a curve who represents the false positive rate (FPR) against the true positive rate (TPR) for all possible cutoffs (Kürüm, Yildirak and Weber, 2011). The true positive rate is the number of good borrowers correctly classified as good borrowers. This rate is also referred as the models Sensitivity. The false positive rate is the number of bad borrowers incorrectly classified as good (1 – Specificity) (Kennedy, 2013).

$$\text{True positive rate} = \text{Sensitivity} = \frac{TP}{TP+FN} = \frac{Corrctly\ predicted\ to\ be\ good}{All\ actual\ good\ borrowers}$$

$$\text{False positive rate} = 1 - \text{Specificity} = 1 - \frac{TN}{TN+FP} = 1 - \frac{Correctly\ \ predicteded\ to\ be\ bad}{All\ actual\ bad\ borrowers}$$

The ROC curve starts in the bottom left corner with a cutoff of zero. Here, every customer is classified as default. Both the true positive rate and the false positive rate are zero, hence everyone is classified as default. The ROC curve ends in the top right corner where both TPR and FPR are one. Here, everyone is classified as good. In the figure below, the black ROC curve is a 45° line which does not predict anything (Kürüm, Yildirak and Weber, 2011). A good model has a high level of sensitivity and specificity, and thus a ROC curve close to the top left corner. In the figure below, the green curve has the best predictive power.

Figure 4.7.1: Example of ROC curve



When comparing ROC curves, one usual calculates the area under the ROC curve (AUROC). AUROC has a value between 1 and 0,5, where 1 is a perfect model. The AUROC can be calculated using the Gini measure (Kennedy, 2013)

$$\text{AUROC} \approx \frac{1 + \text{Gini}}{2}$$

Where the Gini coefficient is defined as (Anderson, 2007):

$$\text{Gini} = 1 - \sum_{i=1}^{n}((cpN_i + cpN_{i-1})((cpP_i - cpP_{i-1}))$$

Where cpP is the cumulative % of good borrowers and cpN is the cumulative % of bad borrowers for a given score.

4.7.2 Brier score

Additional to the AUROC statistics, the Brier score is also widely used as a performance measure when evaluating credit models. While the AUROC asses the discriminatory power of the model, the Brier score asses the accuracy of each predicted probability (Lessmann et al., 2015). If all predicted probabilities of default, for some reason, where doubled for all datapoints in the test data, the AUROC value would still be the same. Brier score gives an insight on how close the model is predicting the true outcome. Therefore, the Brier score is a useful measure to include. The formula for the Brier score is:

$$\text{BS} = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2$$

Where N is the number the number of observations to be evaluated, p_i is the probability of observation *i* is good, and y_i is the actual binary outcome. The Brier score is also referred to as

the mean squared error of the probability estimates. Lower Brier score implies a more accurate model.

5. Feature engineering and variable selection

5.1 Coarse classing

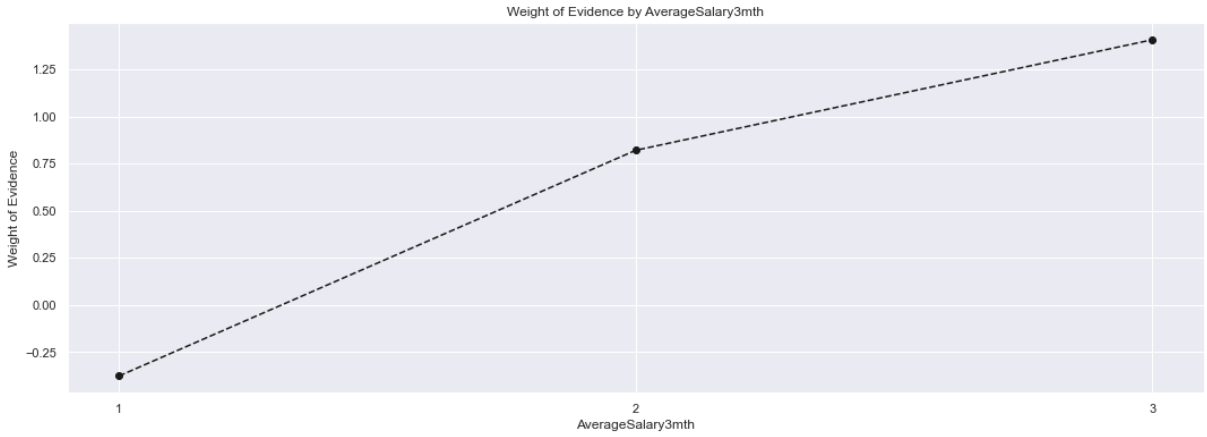
This process consists of splitting the continuous variables into bins by using WOE. Due to the confidentiality agreement with the bank, the bins created at this stage will be named with numbers instead of the real name of the bin. For the discrete variables, the WOE for each group will not be revealed because of confidentiality agreement. All the final bins have at least 5 % of the population in the training data, and observations of both outcomes. The bins created will be transformed to dummy variables. The dummy variable with the lowest WOE will be the reference category. The reference category is not included in the regression to avoid the dummy variable trap.

5.1.1 Income variables

There are two variables for income in this dataset. The first one is salary for three months back in time. This is salary received from an employer. The other income variable is "Payment". Also, this variable is three months back in time. This variable includes all payments to the borrowers account in this bank. It excludes interest rates and transfers between the borrower's accounts in the bank. It could be salary from employer, transfer from other banks, payment from loan and so on.

In order to make the model more robust towards monthly fluctuation in salary, the variable "AverageSalary3mth" is used. This variable is as the name implies, a three-month average of the salary. This variable would logically have an increasing WOE curve. Increased income would all other equal lead to a better financial condition for the borrower and increase the possibility for the borrower to be able to pay his bills. The average salary variable is separated into the three bins. Bin one is when income is lowest and bin three the highest.

Figure 5.1.1a: WOE for average salary three months



The “Payment” variable consists of some of the same information that the salary variable provides. Therefore, a new variable is created. The salary variable is subtracted from the payment variable. By doing this, the problem of the variables telling some of the same information is removed. Now there are two variables, one for salary, and one for all other income except salary. A three-month average is calculated the same way as for salary. After fine classing, the WOE curve for three-month average “Payment” is increasing as expected. The figure below shows that the variable is separated into four bins. Bin one is the lowest payment income group, bin four the highest.

Figure 5.1.1b: WOE for average payment three months

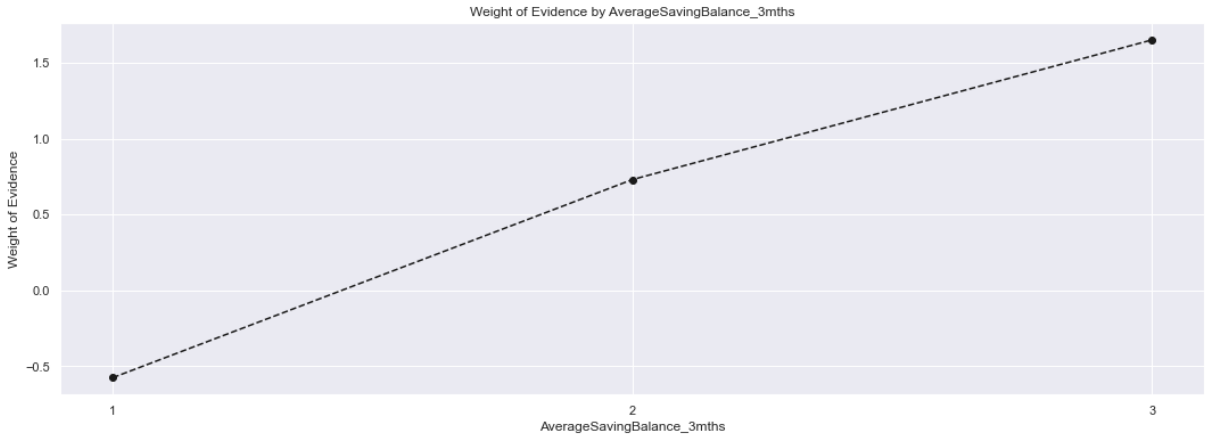
5.1.2 Savings variables

There are three variables for savings in the dataset. A dummy variable who takes the value of one if a borrower has a savings agreement and zero else. The second variable is the length of this savings agreement measured in months. These two variables are naturally highly

correlated, and if both variables are included it could cause problems with multicollinearity. Therefor the dummy variable for savings agreement is included, and the length is not.

The third variable is saving balance. This is a variable for the total savings the borrowers have in their savings account. There is information on the saving balance for the last three months. Also, here a three-months average is calculated for a more robust model. The bins after fine classing are illustrated below. The WOE curve is increasing as expected. Bin one is the group for the lowest saving balance, and bin three is the highest.

Figure 5.1.2: WOE for average saving balance three months



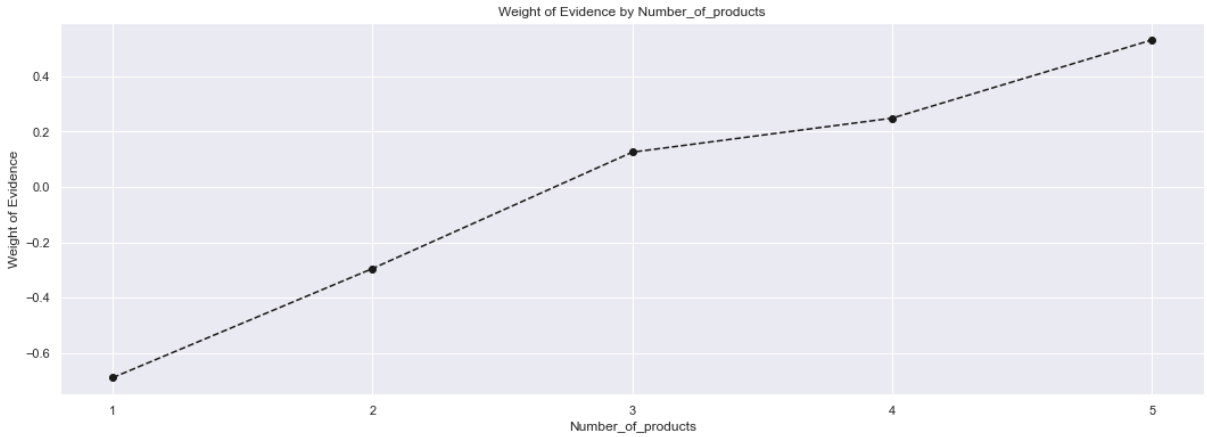
5.1.3 Customer relationship variables

There are several variables with different information regarding the customer relationship to the bank. The variables included in the model are “ActiveOnlineCustomer”, “NumberOfProducts”, “Products”, “LengthCustomerRelationship” and “Transactions”.

“ActiveOnlineCustomer” is a dummy variable equal to one if the borrower has been active on online banking the last month, and zero else.

“NumberOfProducts” is a variable that represents the number of all types of loans a borrower has in the bank. This is the sum of car loans, mortgages, and unsecured credit for a borrower. This variable is split into five bins, and the WOE curve is increasing with number of products. Bin one has the lowest amount of product, bin five the highest. The WOE curve for this variable requires some discussion. If borrowers have a high number of products, one could argue that these are risky borrowers who requires a lot of credit. However, on the other side, a borrower who wants more products can also be a “better” borrower than a borrower with for example only a consumer loan. When banks decide whether to grant credit or not, they check their credit history. If a borrower has repaid all their loans and want new credit it is easier to accept their applications. Considering this last argument, an increasing WOE curve is logical.

Figure 5.1.3a: WOE for number of products



In the original dataset there are three variables for how many of the products car loans, mortgages, and unsecured credit a borrower has. To increase the explanatory power a new variable is created by interacting these three variables. This new variable is called “Products”. This variable has four outcomes. The first outcome is "Only_>1_Unsecured”. If a borrower has more than one unsecured loan like consumer loan and credit card, but no car loan and mortgages, the dummy variable takes the value of one for this variable. The second one is “Only_1_Unsecured”, where the dummy takes the value of one if the borrower only has one unsecured loan in the bank. The third variable is “Unsecured_and_Carloan_notMortgage”, which is the outcome if the borrower has at least one unsecured loan and a car loan, but not a mortgage. The last outcome is “Unsecured_and_Mortgage”. This is the outcome if a borrower has at least one unsecured loan and a mortgage. In the model, these four outcomes or bins, will be included as dummy variables. The dummy variable for “Only_>1_Unsecured” will be used as a reference category, and thus removed from the model. The WOE curve is increasing, and “Only_>1_Unsecured” has the lowest WOE and “Unsecured_and_Mortgage” the highest. This is logical, because if a borrower only has unsecured credits, it is reasonable to assume that this borrower is in a weaker financial position than a borrower who also own their own house with a mortgage.

Table 5.1.3: New variable products

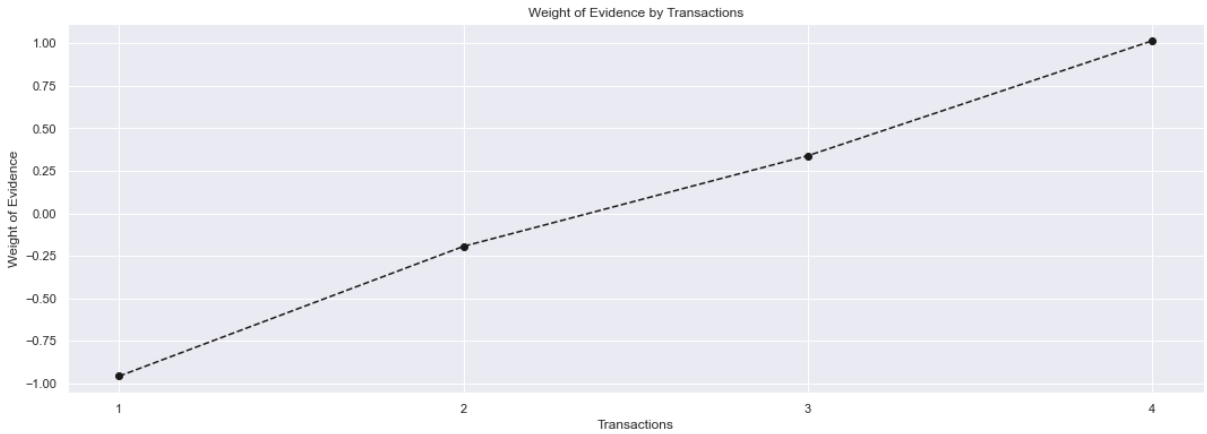
Original variable	New variable
Number of car loans	Only >1Unsecured
Number of mortgages loans	Only_1_Unsecured
Number of unsecured loans	Unsecured and Carloan not Mortgage
	Unsecured and Mortgage

“LengthCustomerRelationship” is a variable who measures how long time the borrower has been a customer at the bank measured in months. After the fine classing, this variable is split into three bins, where group one has the lowest number of months. The WOE curve is increasing, which implies that borrowers who has been a customer for a long time has a lower default rate. Bin one has the lowest number of months, and three the highest.

Figure 5.1.3b: WOE for length of the customer relationship.

“Transactions” is a variable who shows how active the borrower is on his accounts. The variable counts the number of transactions the borrower has on his account within the last month. After fine classing, the variable is separated in to four bins. Bin one has the lowest number of transaction and bin four the highest. As we see from the graph below, the WOE curve is increasing. The more transactions a borrower make, implies a lower probability of default, all other equal.

Figure 5.1.3c: WOE for transactions



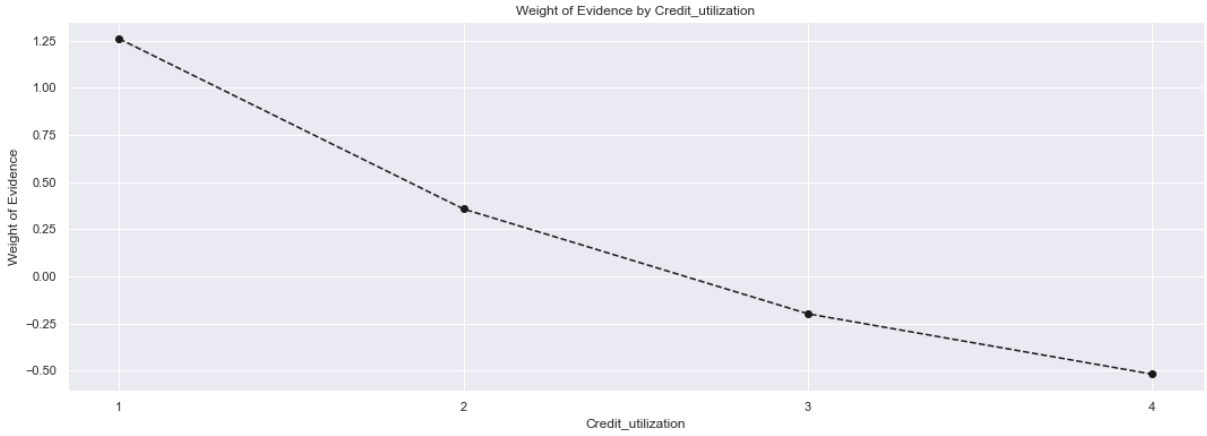
5.1.4 Credit variables

Among the credit variables there is a variable for granted credit both for consumer loan and a variable for other unsecured loans (credit card and account credit). There is a variable for used

credit, the utilization rate. There is also a variable for the number for first and second reminders for car loan, mortgage, and unsecured credit the borrower has.

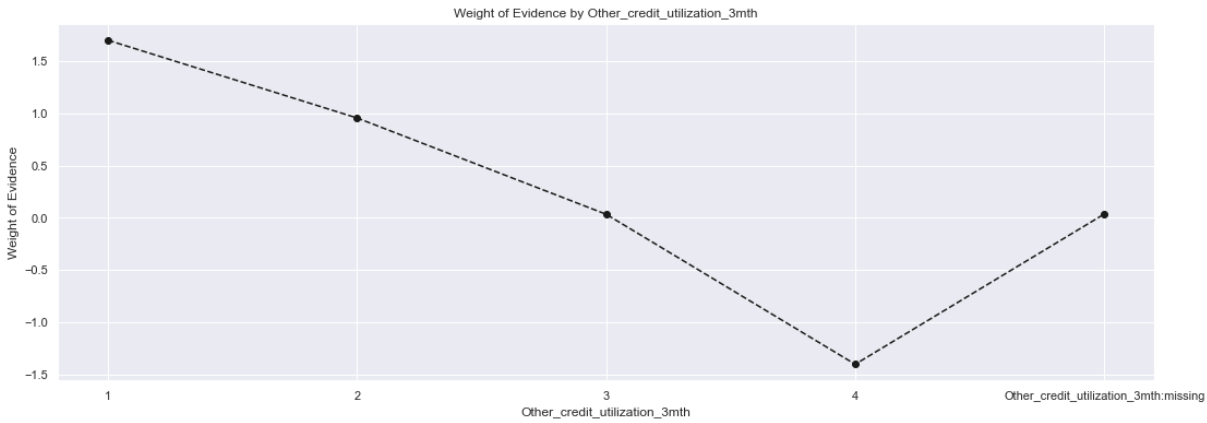
"Credit_utilization" is the utilization rate for consumer loan only. It is calculated by dividing used credit by total granted credit for this product. A high utilization means that there is a high amount left to repay to the bank. The WOE of this variable is expected to decrease as the credit utilization increase. This means that borrowers who has a large amount left to repay will be likelier to default than those who has a small amount left. After fine classing this variable is separated into to four bins, where bin one has the lowest utilization, and bin four the highest. A borrower with a high utilization, and thereby has more left to repay their consumer loan, is more risker than one who has less to repay.

Figure 5.1.4a: WOE for credit utilization



"Other_credit_utilization_3mth" is a variable for all other unsecured credit. Consumer loan is excluded. Examples of unsecured credit included in this variable is credit-card and account credit. The utilization rate is calculated the same way as for "Credit_utilization". However, for this variable, a three-month average is calculated. There is a high number of missing values for this variable. This is because there are many borrowers who do not have these credit products. Therefore, these are not really missing, but in the graph below the group is called missing. When fine classing is done, this variable is separated into four bins plus one group for the missing. Bin one has the lowest credit utilization, and bin four the highest.

Figure 5.1.4b: WOE for other credit utilization

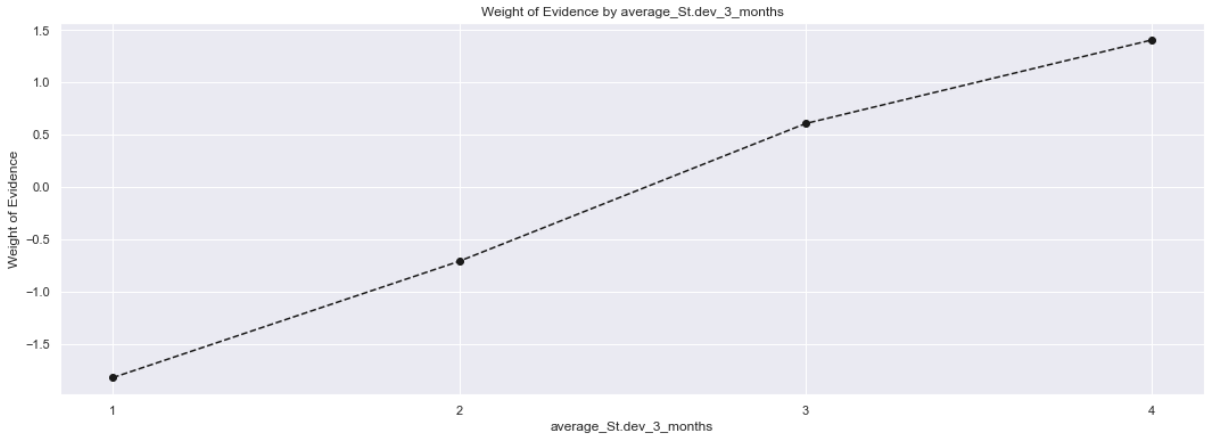


In the data there are variables for first and second reminder for all credit types. In the original data provided by the bank, there is a variable who interacts these variables to one variable, “Group reminders”. This variable originally had three values; “No reminders”, “first reminder, but not second” and “has second reminder”. Of these outcomes, “No reminders” have the highest WOE and having “has second reminder” the lowest. These variables are transformed to dummy variables, and the dummy for having two reminders is the reference category and thus removed.

5.1.5 Daily balance data

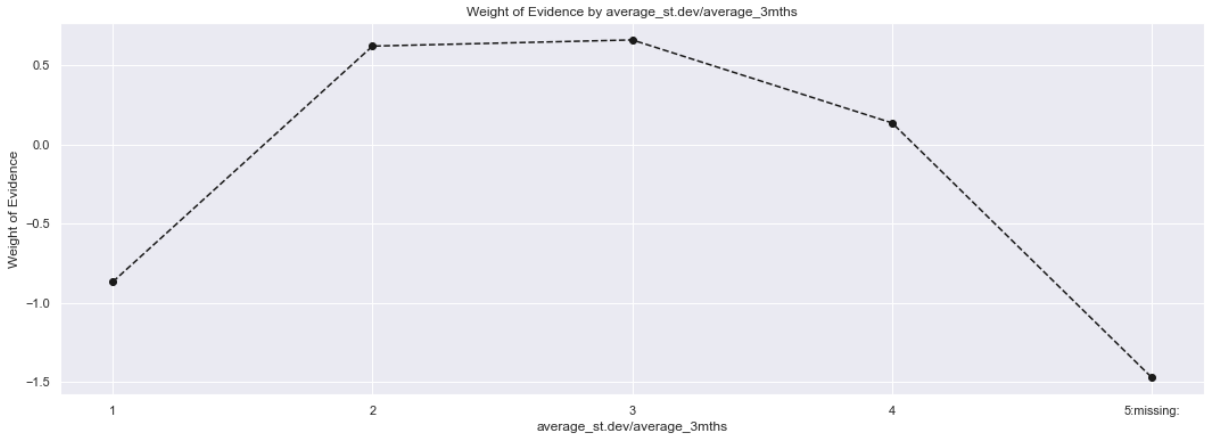
From the daily balance dataset, a new variable “St.dev” is created. This variable is the standard deviation calculated from the borrower’s positive balance. It is calculated month by month, and therefore possible to connect to the other dataset. Standard deviation is here a measure on the variation of the daily balance. A high standard deviation implies a borrower who has large fluctuation on his account during a month. Also, for this variable, an average of the last three months is created. This variable is separated into five bins. Bin one has the lowest standard deviation and bin five the highest. The WOE curve is increasing, which is logical. A high standard deviation implies high fluctuations, but also that the borrowers have high sum of money in their account in the first place.

Figure 5.1.5a: WOE for three-month average of standard deviation



To further investigate the relationship between the fluctuations in the daily balance of the borrowers, a new variable is created. This variable is the average standard deviation divided by the average balance for the same month. A value of one means that the standard deviation is equal the average balance of the borrower's accounts. A borrower who has little variations in his balance has a "St.dev/average" which is lower than one. After fine classing, this variable is separated into five bins, where one is for missing variables. Bin one has the lowest value (except the missing), and bin four the highest. The WOE curve is concave. Having a high or low value of this variable yields higher risk all other equal. This means that having either small fluctuations in the daily balance relative to the average value or high fluctuation implies higher risk. In the first bin there is many who has a "St.dev" of zero. These borrowers may use other banks as their main bank, and thus gives little information. Besides this, the WOE curve is logical. Higher variation in the daily balance implies a lower WOE and all other equal higher risk of default.

Figure 5.1.5b: WOE for three-month average of st.dev/average



5.1.6 Variable interactions

In addition to the variables already mentioned some variable interactions are included. By adding an interacting term, it is possible to capture a relationship between two variables. The two added variable is one interaction between two continuous variables, "LengthCustomerRelationship*Credit_utilization", and one interaction between a dummy variable and a continuous variable, "St.dev*ActiveOnlineCustomer".

With "LengthCustomerRelationship*Credit_utilization" the goal is to investigate how the credit utilization depends on how long time a borrower has been a customer in the bank.

There is a high number off borrowers who has a "St.dev" of zero. By adding the interaction term "St.dev*ActiveOnlineCustomer", it is possible to investigate how "St.dev" affects the dependent variable if only the borrowers who is active online customers is included.

However, when calculating correlation between the variables, both variables are highly correlated with respectively "LengthCustomerRelationship" and "St.dev". Therefore, these interaction terms are removed before model training.

5.2 Variable selection

After the binning process there is 14 variables left which is not highly correlated to another variable nor have an IV below 0,02. When using the Akaike information criteria, I start by including the most informative variables and then add the other most informative variables one by one. The model with the lowest AIC is when all variables is included. Therefore, when training the models, we do not remove any of the 14 variables.

To investigate if the including of all variables leads to a poor out-of-sample performance (overfitting), the models are also trained one time with a dataset where variables are chosen after backwards elimination. Backwards elimination starts with including all potential features in the regression, before a removal of the features which is least significant (Siddiqi, 2017). All variables with a p-value higher than 0,05 is removed.

After this stage, eight variables are not statistically significant and thus removed from this dataset. These are: "NumberOfProducts", "Average_st.dev/snitt_3mths", "Average_st.dev_3mths", "AverageSavingBalance_3mths", "Average_payment_3mths", "AverageSalary3mths", "SavingsAgreement", and "ActiveOnlineCustomer".

Summarized, after the variable selection the models are trained and tested using four datasets. The number of variables in the table is after the categorical variables are transformed to dummy variables:

Table 5.2: Overview over the different datasets

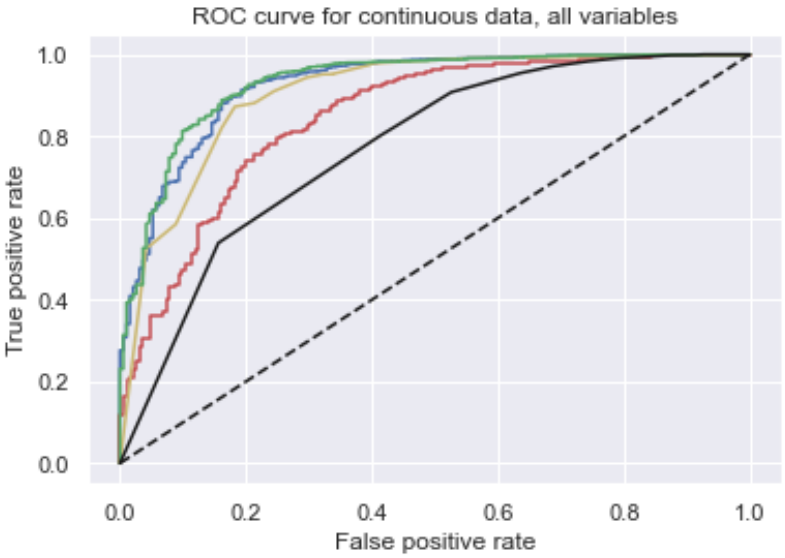
Dataset	Variables (Including dummies)	Discrete variables	Continuous variables
Continuous dataset -all variables	17	Transformed to dummy variables	Kept in its raw form
Dummy dataset -all variables	37	Transformed to dummy variables	Separated into bins using WOE.
Continuous dataset -only significant variables	9	Transformed to dummy variables	Kept in its raw form
Dummy dataset -only significant variables	14	Transformed to dummy variables	Separated into bins using WOE.

6. Results

When testing the models, the models are tested on the out-of-sample data. This gives us an unbiased indicator of how well the models has performed. All models are tested using the continuous data and the dummy data, where I test with all variables and the significant variables for both.

6.1 Continuous data

Figure 6.1.1: ROC curve for continuous dataset, all variables



Model	Color
Logistic regression	
K-NN	
Decision tree	
Gradient booster	
Random forest	

Figure 6.1.2: ROC curve for continuous dataset, significant variables

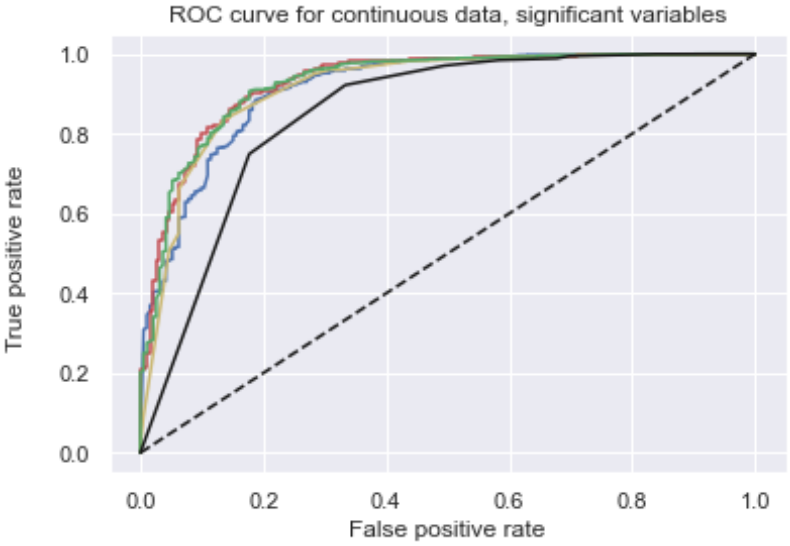


Table 6.1: Results for the continuous dataset

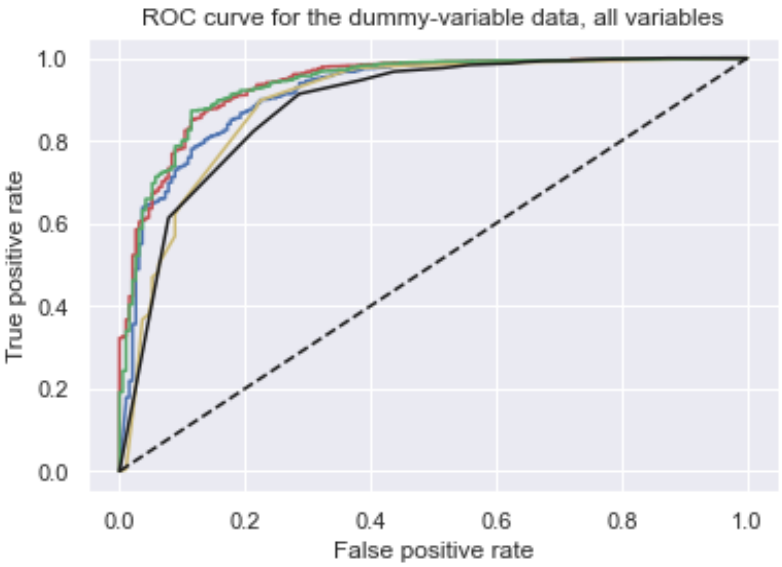
	All variables		Only significant variables	
Model	**AUROC**	**Brier Score**	**AUROC**	**Brier Score**
Logistic regression	0.8467	0.0567	0.9296	0.0381
k-NN	0.7728	0.0601	0.851	0.0486
Decision tree	0.9041	0.0418	0.9163	0.0408
Random forest	0.9254	0.0399	0.9139	0.04
Gradient Boosting Classifier	0.9318	0.039	0.9286	0.0384

For the continuous dataset, where the continuous variables are not engineered except treating missing values logistic regression and k-NN are the models who suffers the most from adding insignificant variables. These models perform poorer both regarding AUROC and Brier score when using all variables. Looking at these results, logistic regression performs significantly worse than the tree models, especially the more complex models when all variables are included. This could be an indicator of that this model does not capture the non-linear effects in the dataset as well as the tree-models. The tree-based models select the best variables when building the tree, which means that they find the best variables themselves. This could be an explanation of why these performs almost the same for both datasets. The logistic regression has not this feature.

When only the significant variables are used, the logistic regression and gradient booster has the highest score and performs almost the same, both when measuring Brier score and AUROC. This implies that among the insignificant variables, there are some relationships that logistic regression does not capture.

6.2 Dummy variables

Figure 6.2.1: ROC curve for the dummy variable dataset, all variables



Model	Color
Logistic regression	
K-NN	
Decision tree	
Gradient booster	
Random forest	

Figure 6.2.1: ROC curve for the dummy variable dataset, only significant variables

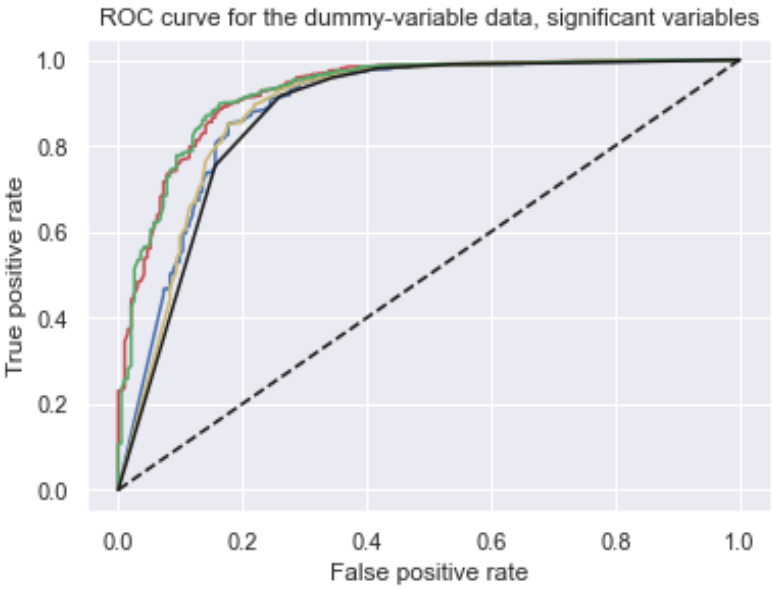


Table 6.2: Results for the dummy dataset

	All variables		Only significant variables	
Model	**AUROC**	**Brier Score**	**AUROC**	**Brier Score**
Logistic regression	0. 9372	0.0377	0.9273	0.0388
k-NN	0.8838	0.047	0.8735	0.0444
Decision tree	0.8929	0.0431	0.8866	0.0416
Random forest	0.9173	0.0418	0.8865	0.0429
Gradient Boosting Classifier	0.9361	0.0379	0.9273	0.039

When using the dataset where the continuous variables are spitted into intervals and each interval are transformed into dummy variables, all the models perform almost the same for both datasets. Also, for the dummy variable dataset, the gradient booster and logistic regression are the best models with the highest AUROC and lowest Brier score.

Summarized for both the continuous data and dummy variables dataset, gradient booster and logistic regression are the most accurate models. The logistic regression manages to capture the same relationship that the more complex model gradient booster does, except for the continuous data where all variables are included. Random forest does almost perform as good as these two models when using all variables in the continuous dataset. As expected, the less complex model's decision trees and k-NN has the lowest performance, but their score is not bad. K-NN performs best when all variables are dummy variables. This is expected because k-NN performs best when the data is preprocessed. These two models manage to capture much of the relationship of the dataset.

6.3 Economic impact of the different models

To illustrate the main finings in a real-world business problem, a model who shows the economics behind the problem is created. The purpose of this model is to highlight the distinction in results when using different models, measured in NOK.

In this model 1.000 datapoints is randomly drawn from the test data. Their probability of being a good borrower (not default) is estimated and based on this probability they are approved for a loan or not. We assume that every borrower who is approved for a loan, is granted a credit of 100.000 NOK which should be repaid in one years' time. The one-year interest rate is 10 %.

The income for the bank when a borrower who fulfill its obligation is the interest rate income 10.000 NOK (100.000 NOK * 0,1).

For simplicity, the model assumes that the only costs for the lender occurs when a borrower defaults. The lender could sell the defaulted loan to credit institution for 50 % of the principal (interest rates is not included). For a defaulted consumer loan of 100.000 NOK, the bank has a loss of 50.000 NOK. The rate of 50 % could be a little high. The market price for defaulted loans could be affected by many factors. These could be the risk profile of the loan, the supply side, and the demand side of these loans. Therefore, calculation where the lender can sell defaulted loan for 40 % and 30 % of the principal.

This model is very simplified. In real life, there are other costs related to the scenario described above. The purpose of this model is to give an economic description of how the ability to separate good borrowers from bad borrowers would be measured in NOK.

This model looks at two thresholds for when the lender classifies the borrower as predicted good or bad (non-default and default). The borrowers who are classified as bad are not offered the 100.000 NOK consumer loan. The first threshold is: reject all who has a probability of being good below 85 %, and the second; reject all who has a probability of being good below 95 %.

The model uses the dummy data using significant variables to evaluate k-NN, logistic regression and gradient booster. K-NN is included because it performs the worst of the models, gradient booster and logistic regression performs almost the same. Using the dummy variable dataset with only significant variables, k-NN had a AUROC of about 0,87, logistic regression and gradient booster both had an AUROC of around 0,93.

By comparing one "strict" threshold and one less "strict" threshold it provides us with an intuitive understanding of how the difference between the models is measured in profit. When evaluation the "strict" threshold, another random sample is drawn.

Table 6.3.1: 85 % threshold

Model	Logistic regression	Gradient booster	K-NN
Offered loans	892	891	859
Defaulted loans	24	24	22
Profit (70 % loss)	NOK 7 000 000	NOK 6 990 000	NOK 6 830 000
Profit (60 % loss)	NOK 7 240 000	NOK 7 230 000	NOK 7 050 000
Profit (50 % loss)	NOK 7 480 000	NOK 7 470 000	NOK 7 270 000

The table above shows the profit when using the three models where only borrowers who have a predicted probability of being good higher than 85 % is granted the consumer loan. The logistic regression yields the highest profit, but only 10.000 NOK higher than for the gradient booster when the bank suffers 50 % loss for defaulted loans. If the k-NN were used in this case, the profit would be 210.000 NOK less than the best model, logistic regression.

Table 6.3.2: 95 % threshold

Model	Logistic regression	Gradient booster	K-NN
Offered loans	747	724	706
Defaulted loans	10	9	14

Profit (70 % loss)	NOK 6 670 000	NOK 6 520 000	NOK 5 940 000
Profit (60 % loss)	NOK 6 770 000	NOK 6 610 000	NOK 6 080 000
Profit (50 % loss)	NOK 6 870 000	NOK 6 700 000	NOK 6 220 000

The table above shows the profit when the threshold is a little stricter. Here, a new random sample of borrowers from the test data. For every model, fewer loans are offered. For this threshold, logistic regression has a higher profit than gradient booster. Also in this case, k-NN is the worst model. The difference between the best model and the worst model is 650.000 NOK, a much bigger difference than in the case with 85 % threshold.

Summarizing the results from the different scenarios, we observe that there is a big difference between the best and worst model for the 95 % threshold. There is a notable difference in profit, even with only 1.000 datapoints included in this model. Using a better model would be of great benefit for the lender. Logistic regression gives the highest profit, and the biggest difference is for the 95% threshold.

6.4 Discussion

When using both Brier score and AUROC, it is possible to rate the model's discriminatory power and the accuracy of predicted probabilities (Lessmann et al., 2015). By using both measures, it becomes possible to draw a conclusion on the credit models' performance with more certainty than using only one measure. When looking at the results, it is some connection between the Brier score and AUROC. The model who has the lowest AUROC also has the Brier score overall (k-NN).

The answer to the question of which model is the best is not straight forward. Overall, gradient booster and logistic regression performs the best both regarding AUROC and Brier score. Random forest is less accurate then the two models, but it is not a big difference. Gradient booster has an impressive performance for all dataset, and it manage to predict the behavior of the borrowers in a good way. But the logistic regression also has a good performance. Even though logistic regression is a less complex model, it manages to capture the same dynamics and relationship as the more complex models do. One disadvantage of logistic regression relative to the tree models is the assumption of a linear relationship between the target variable and log the odds (the logit transformation). After reviewing these results, logistic regression performs just as good as gradient booster except when all variables are included, and continues variables are in their raw form. It is likely that poorer performance on this dataset is a combination of that some of these variables do not have a linear

relationship with the log odds. When splitting the continuous variables into bins, the logistic regression manages to capture possible non-linearity. The gradient booster, by creating a forest of weak learners who corrects its previous mistakes, manage to capture both linear and non-linear relationships.

Looking at the model who illustrates the result on the bottom line for the lender, the profit of the two models is almost equal for the less strict model. But for a strict model, logistic regression yields the highest profit. While gradient booster is hard to interpret, logistic regression gives much more room for interpretation. One could investigate the feature importance output the gradient booster offers, but the possible interpretation of the coefficients of the logistic regression is much easier to explain. It is also a big benefit to have the opportunity to transform the coefficients to a credit scorecard like the one presented in the introduction. Using a scorecard like this, it is easier to explain the results from the analysis to risk managers. The lack of interpretability and the fact that training the gradient booster is time consuming, are huge disadvantages of this model. I therefor conclude that the logistic regression has the best overall attributes after using the data from the bank.

7. Conclusion

In this thesis, I have compared machine learning techniques to logistic regression. Different behavioral credit scoring models based on consumer loan borrowers has been created to compare the different algorithms. The machine learning models used to compare with logistic regression is k-nearest neighbor, decision trees, random forest, and gradient booster. I find that gradient booster and logistic regression is the most accurate models when it comes to behavioral forecasting. This is done by comparing the model's discriminatory ability (AUROC) and predicted probabilities accuracy (Brier score). Logistic regression is the best overall model, because of its interpretability and high accuracy. Therefore, this thesis concludes that logistic regression still is very accurate and have some other big benefits and thus is recommended as the preferred model to the bank.

8 References

Anderson, R. (2007). *The Credit Scoring Toolkit. Theory and Practice for Retail Credit Risk Management and Decision Automation.* New York: Oxford university press.

Athey, S. (2018). The Impact of Machine Learning on Economics, in Agrawal, A. Gans, J. and Goldfarb, A. (eds) *The Economics of Artificial Intelligence: An Agenda.* Chicago: University of Chicago Press. pp. 507-547

Athey, S. and Imbens, G. W. (2019*).* Machine Learning Methods That Economists Should Know About. *Annual Review of Economics.* 11(August 2019). pp. 685-725. doi: 10.1146/annurev-economics-080217-053433

Baesens, B. et al. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society,* 54 (6), pp. 627-635. doi: 10.1057/palgrave.jors.2601545

Bolton, C. (2009*). Logistic regression and its application in credit scoring*. Dissertation (MSc). University of Pretoria. URI: http://hdl.handle.net/2263/27333 (Accessed: 15.04.2021)

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(October 2001), pp. 5–32. doi: 10.1023/A:1010933404324

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

Crook, J. N., Edelman, D. B. and Thomas, L.C. (2007) Recent developments in consumer credit risk assessment. *European Journal of Operational Research*. 183 (3), pp. 1447-1465. doi: 10.1016/j.ejor.2006.09.100

Finanstilsynet (2020 a). *Lavere volum og høyere mislighold i forbrukslånsmarkedet,* Available at: https://www.finanstilsynet.no/nyhetsarkiv/nyheter/2020/lavere-volum-og-hoyre-mislighold-i-forbrukslansmarkedet/ (Accessed: 31.05.2021)

Finanstilsynet (2020 b) *Identifisering av misleghaldne engasjement.* Available at: https://www.finanstilsynet.no/nyhetsarkiv/rundskriv/2020/identifisering-av-misleghaldne-engasjement/. (Accessed: 31.05.2021).

Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. *European Journal of Operational Research*, 210 (2), pp. 368-378. doi: 10.1016/j.ejor.2010.09.029

Florez-Lopez, R. and Ramon-Jeronimo, J. (2015) Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal. *Expert Systems With Applications.* 42 (13), pp. 5737-5753. doi: 10.1016/j.eswa.2015.02.042

Hastie, T. Tibshirani, R. and Friedman, J. H. (2009). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Second Edition. New York: Springer, NY. Available at: https://web.stanford.edu/~hastie/Papers/ESLII.pdf (Accessed: 28 April 2021).

Iskhakov, F. Rust, J. and Schjerning, B. (2020) Machine learning and structural econometrics: contrasts and synergies. *The Econometrics Journal*, 23 (3), pp. S81–S124, doi: 10.1093/ectj/utaa019

Kennedy, K. (2013). *Credit scoring using machine learning*. Doctoral thesis. Technological University. Dublin. Available at: https://arrow.tudublin.ie/sciendoc/137/ (Accessed: 10 March 2021)

Konishi, S. and Kitagawa, G., (2008). *Information Criteria and Statistical Modeling.* New York: Springer Science + Business Media, LLC. Available at: https://link.springer.com/content/pdf/10.1007%2F978-0-387-71887-3.pdf. (Accessed: 15 May 2021)

Kruppa, J. et al., (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications,* 40 (13), pp. 5125-5131. doi: 10.1016/j.eswa.2013.03.019

Kürüm, E., Yildirak, K. and Weber, G-W. (2011) A classification problem of credit risk rating investigated and solved by optimisation of the ROC curve. *Central European Journal of Operations Research.* 20 (September 2012), pp. 529–557. doi: 10.1007/s10100-011-0224-5

Lessmann, S. et al. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*. 247 (1), pp. 124-136. doi: 10.1016/j.ejor.2015.05.030

Liu, W. Fan, H. and Xia, M. (2021). Step-wise multi-grained augmented gradient boosting decision trees for credit scoring. *Engineering Applications of Artificial Intelligence*. 97(January 2021), doi: 10.1016/j.engappai.2020.104036

Mullainathan, S. and Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives.* 31 (2), pp. 87–106. doi: 10.1257/jep.31.2.87

Müller, A. C. and Guido, S. (2017). *Introduction to Machine Learning with Python*. Sebastopol, CA: O'Reilly Media, Inc.

Poppe, C (2017) *Usikret Kredit - et samfunnsproblem?* SIFO report. Oslo: Forbruksforskningsinstituttet SIFO

Siddiqi, N. (2017*). Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards (Wiley and SAS Business Series)* 2nd Edition. Hoboken, New Jersey: John Wiley and Sons, Inc.