Norwegian School of Economics

Bergen, Spring 2020

Predicting Defaults in the Automotive Credit Industry

An Empircial Study Using Machine Learning Techniques Predicting Loan Defaults

Petter Telle Bøe

Supervisor: Geir Drage Berentsen

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This master thesis was written as part of the Master of Science in Economics and Business Administration, with a major in Business Analytics at the Norwegian School of Economics(NHH).

This master thesis has been a very satisfying experience working with a broad and high-dimensional data set, compared to the two years of my major in BAN. In addition I have developed an greater insight into the many different machine learning techniques and the potential of the credit industry.

I would like to express my gratitude to Geir Drage Berentsen for beeing regulary in contact, and the very supportive, while keeping me ahead with the deadline through our the semester.

Abstract

This master thesis explore the potential of Machine Learning techniques in predicting default of vehicle loan applicants. Usually, banks or other financial institutions utilize the Logistic Regression algorithm to support their decisions-making process, however more advanced methods has been proven to advance in classifying default predictions. The data set applied in this are collected from several institutions, contained contract information, historical credit information and status, and demografical information of more than 240 000 granted loan applicants.

The results from four different machine learning techniques; Random Forest, Gradient Boostin Machines, Support Vector Machines and Neural Networks, were compared to the benchmark model; Logistic Regression. From the study, the Neural Network were found marginally better than the Logistic regression. Notably, all models were trained and tested on identical data set, however separated the fitting, validation and the testing in three data sets with similar features. However, due to time- and computational constraints, the models was not fully exploited in terms of tuning the hyperparameters.

The best performing model, Neural Network, achived an AUC of 0.6349, followed closely by the Logistic Regression with an AUC of 0.6325. Based on the performance and knowledge of the models, a conclusion that the Logistic Regression is the best, however the Neural Network has the best potential in towards future research due when data qualty.

Contents

List of Figures

List of Tables

1 Introduction

Automobiles such as cars and light trucks are the most commonly held non-financial asset which in recent decades has increased significantly much due to the economic growth. In fact, 268 million new registered motor vehicle owners were found in 2019 in the USA and around 70.5 million new commercial vehicles were produced worldwide (SRD, 2020). In the five years to 2019, sustainable macroeconomic growth has led to an increased demand of lending, specially in the Financing sectors. Additionally, interest rates have decreased, creating cheaper loans, which could be one of the reasons that around 85 percent of new passenger vehicles on the road is financed through a loan or lease (Experian, 2020). The large amount of vehicle financed through loans has in fact resulted in a tremendously large amount of outstanding debt, which today stand out with 1.3 trillions of dollars(SRD, 2020).

The large amount of debt outstanding for vehicles is held mostly by banks. These loans mainly originates from direct lending in banks or indirectly from the vehicle dealer(Agarwal, Ambrose, & Chomsisengphet, 2008). In addition to a competitive market for automobile credit, a significant drop in the interest rates in the past years has made the prediction of good and bad borrowers important. Given that a buyer refuses to pay off the debt, the banks can lose a great deal of money as, with time, the valuation of the vehicles also loses considerable value. According to the International Monetary Fund (IMF), a loan can be categorized as default if:

- "Loan installments of principal and interest are at least 90 days due, and the lender no longer believes the borrowers will honor their debt obligations. In this case, the loan is written off as a bad debt in the lender's books of accounts".
- "Ninety (90) days' worth of interest payments are capitalized, refinanced, or delayed due to changes in the loan agreement."
- "Payments of principal and interest are less than 90 days overdue, and there are reasons to doubt that the borrower will not pay the outstanding loan in full."

When a lender obtains a substantial proportion of their unpaid loans as default, the financial performance might be affected, due to loss in interests received from deposits(CFI,

2020). Except for the time period around the financial crisis in 2007-2008, the non-performing ratio has remained quite stable across all countries(CFI, 2020). Despite emergence of a financial recession, banks have often varying strategies to handle a defaulted loan, however most of the following options may trigger transaction costs(CFI, 2020). - The lender will take possession of the motor vehicle and sell it off to recover any amounts owed by the borrower. - The lender may opt to sell the defaulted loans to collection agencies or outside investors to get rid of the risky assets from their balance sheet. - Alternatively, the lender can engage a collection agency to enforce the recovery of a defaulted loan in exchange for a percentage of the amount recovered.

Banks can increase their profit and reduce the portion of defaulted borrowers by creating systems to decide which are good and bad customers. As the interest rates drop, the revenue per customer is significantly lower compared to a bad borrower's impact on their profit. Therefore, the biggest concern for banks is estimating credit risk (Bekhet & Eletter, 2014). As a matter of fact, evaluation of loan applications tends to be assessed in a subjective nature. For example, Jordanian banks manually reviewing the applications(Bekhet & Eletter, 2014). Following this approach includes personal insights, knowledge and intuition which may impose bias. This method is usually replaced in banks by credit scoring models or a combination of both to make proper decision making. In fact, risk estimation is a challenge and a major factor contributing to financial decision. The inability to determine risk accurately has an adverse effect on the credit management(Bekhet & Eletter, 2014). In addition, risk estimation affects more than a loss given default of borrowers, but also the capacity of liquidity requirements and risk of losing potentially profitable customers.

Credit scoring is a collection of decision models and methods giving support to the banks when offering credit to applicants(Bekhet & Eletter, 2014). In other words, helping managers in the financial decision-making process towards deciding to accept or reject applicants. According to Chen and Huang (2003) credit scoring models are used on decisions related to credit admission evaluations due to the rapid development in the credit industry. Developing such a model is built on historical information of defaulted and non-defaulted applicants(Bekhet & Eletter, 2014). These models are developed with respect to the applicants

characteristics such as age, income, marital status, credit history, etc.(Bekhet & Eletter, 2014). Credit scoring models seek to evaluate the applicant's capacity to meet financial commitments by determining lenders vulnerability. Applicants who find a high probability to satisfy the financial obligations will be granted a loan and vise versa. These model are built on a statistical technique usually Logistic regression or Linear Discriminant Analysis supporting the decisions of the good and bad applicants.

The aim of this thesis is to investigate the potential of Machine Learning algorithms in modelling Probabilities of Default(PD) within the automobile credit industry. Looking at different classification techniques, the goal is to seek insights into how these methods are able to distinquish between good and bad applicants. Logistic regression has been one of the most used algorithms to predict default, and this method is easy to interpret. However, due to the significance of credit risk, several studies have proposed banks to embrace additional data mining tools to improve their risk assessment (Chen & Huang, 2003), which will be evaluated in this thesis. In the next section, a brief overview of previous studies of PD modelling will be reviewed.

1.1 Literature review

Probability of default modelling is a well-researched topic and continues to attract much interest. Within PD modelling, there exist number of papers studying the growth, applications, and evaluations of predictive models used for supporting decisions within the credit industry(Lessmann, Baesens, Seow, & Thomas, 2015). In 1998 Altman and Saunders (1998) published an overview of credit risk modelling for the last 20 years. In addition to their key findings, the authors point out that credit risk modelling has evolved drastically for the past 20 years due to new emerging statistical techniques(Altman & Saunders, 1998). Later, Swedish researchers published an extension to Altman and Saunders work presenting a further development of credit risk modelling(Hao, Alam, & Carling, 2010). This work identifies more than 1000 articles on this topic, finding logistic regression and discriminant analysis as the most widely used methods for constructing scoring systems. However, they also identified studies that proposed the use of new advanced models that had been developed more recently.

These models are typically based on more information about the borrowers. Lastly they pointed out that for the past ten years the current attention of the papers had moved from static individual-level models towards more dynamic modelling (Hao et al., 2010), similar to models proposed by Altman and Saunders. These studies seems to have initiated some interest in more advanced Machine Learning algorithms for PD since plenty of other research paper has been published in the following years. The next paragraphs will present some of these studies and finally present the overview of the models that standout.

Lessmann et al (2015) and kruppa et al(2013) compares several classification algorithms for predicting the probabilities of default. Lessmann et al investigate the overall model performance using several datasets, and examine the predicting performance in each case. The conclusion from this research recommends the Random Forests(RF) model as a benchmark model because of its effectiveness, precision, and its interpretability. Although the authors call attention to the well-performing Neural Networks(NN), advise future studies to apply the NN models because of the potentially good predictive power(Lessmann et al., 2015). Kruppa et al (2013) study and compare multiple classification algorithms such as the Random forest, logistic regression and k-nearest Neighbour. From their research RF outperforms the Logistic regression(Kruppa, Schwarz, Arminger, & Ziegler, 2013).

Agrawal et al (2014) study the impact of contract-specific variables as predictors in commercial vehicle loans. In their research, applying a logistic regression model for predicting default, around 11 out of 17 contract-specific variables where identified to provide additional assistance for the credit lending institution(Agrawal, Agrawal, & Raizada, 2014). The authors also suggest that contract information could improve the accuracy in more advanced non-linear models. Specifically, the authors suggest the use of Neural Networks as one potential predictive model to improve the performance based on contract information(Agrawal et al., 2014).

Many researchers have been studying the default prediction of social lending in recent years. This concept is often referred to as peer to peer lending(P2P)(Aleum Kim, 2019). Studies of P2P, typically consider comparisons of Logistic regression, decision trees, and a few deep learning methods. More recently, Wang et al.(2018) propose a novel behavioral credit scoring

model predicting a dynamic probability of default. In their study RF is found to be the best performing model predicting defaults. However, PD modelling of P2P transaction does not conclude that one model uniformly outperforming other models. Several studies discover the Logistic regression as the best predictive model(CHEN, 2017; Ge, Feng, Gu, & Zhang, 2017; Li, Hsu, Chen, & Chen, 2016; Lin, Li, & Zheng, 2017), while some researchers discover others more advanced machine learning algorithms which perform well when the data is large and complex. The table below displays the best methods found in each research; Random Forest(RF), Neural Network(NN) and Gradient Boosting Machines(GBM).

Table 1.1: Best Machine learning methods found in research. (NA means the research did not apply the measure)

Author	**Method**	**Accuracy**	**AUC**
Fu (2017)	RF	0.7350	NA
Jiang et al (2017)	RF	0.8600	NA
Huo et al. (2017)	NN	0.8796	NA
Kruppa et al(2013)	RF	NA	0.9590
Wang et al (2018)	RF	NA	0.7510
Ma et al (2018)	GBM	0.8010	NA
Chen and Guestrin (2016)	GBM	NA	0.8304
li et al (2018)	GBM	NA	0.7850

Lastly, the Support Vector machine(SVM) has been studied by several researchers(Crook, Edelman, & Thomas, 2007; Jiang & Zhang, 2017). The method were not the best performing method except for Crook et al(2007). Crook et al (2007) study the customer credit worthiness based on information such as balance sheets, financial ratios and macro-economic indicators(Crook et al., 2007). The study finds that the SVM perform well and accurate.

Keeping the literature in mind, this thesis aims to subsidise the automobile industry's decision making. As for the resent papers presented, tree-based methods seems to be the

most repeated and best performing method predicting default. Keeping that in mind, in this master thesis two tree-based methods, Gradient Boosting and Random Forest, will be introduced. In addition, Support Vector Machines and Neural Networks do not seem to be as much researched, however, these models enhance their capabilities to manage massive and complex data, which appear in our data set with about 39 variables and 240 000 observations. In fact, the data included in this thesis contain almost four times more observations as the average data sets reviewed(Aleum Kim, 2019).

1.2 Structure of thesis

This thesis contain in total 6 chapters. In chapter 2, data applied in addition to the preprocessing and splitting of the data. Chapter 3 present the calculations for estimating models, an introduction and developement of the machine learning techniques applied in this thesis. In chapter 4 the analysis of the models performance will be evaluated. Chapter 5 a brief discussion of the validity of the models, limitations of the thesis and interesting areas of future research. Finally in chapter 6, the conclusion of this thesis will be presented.

2 Methodology

Chapter 3 describes the dataset and the mechanisms which are handled in this thesis Section 3.1 elaborate on the dataset and the given information. Section 3.2. describes how the data has been pre-processed of missing values, feature engineering, etc. Lastly, in section 3.3 an description of the dataset and how the data is split before the analysis is initiated.

2.1 Dataset

The original dataset is collected from the database of Kaggle, containing information of granted automobile loans from 2018(Paul, 2019). The dependent variable is a binary variable, taking the value 1 if the borrower has defaulted, and 0 if not. For the independent variables, the data contains information that can be split in three categories; Demographical data of the borrower, loan information(contract based), and Bureau history. Demographic data are information on age, employment status, etc. Among the Loan variables, the dataset contains information of the granted loan, such as disbursed amount, loan to value ratio, application information, etc. Lastly, Bureau history includes information on the customers risk, numbers of accounts, history of granted loans, etc. In total the dataset include (See Appendix 1 for a full description of the original features)

2.2 Data Preprocessing

Data preprocessing is a major step within the Machine Learning(ML) process. Preprocessing of data is a process where the data will be transformed, or encoded, to bring it in such state that an algorithm can easily interpret the features. Features containing text are not understandable for Machines, and several softwares expect categorical variables in a dummy variable formation such as 0´s and 1´s. Much of the data are collections of data, often records, observations, etc. Transforming features into meaningful and intuitive features could benefit the performance of any model.

2.2.1 Handling Categorical variables

The dataset comes pre-cleaned, but some modification is done. The original data contains 41 variables, and 14 are categorical variables including the response variable, $loan_default$, representing if a borrower has defaulted within the first Equity Monthly Instalment (EMI) on the due date. Further, most of the categorical variables contains two levels, while some of the variables exhibit high cardinality. For the purpose of machine learning techniques, this could lead to critical computational problems. To solve such a problem one-hot encoding is introduced as a response to the problem. One-hot encoding is a technique to create a dummy variable (binary variable) for those categorical feature with more than two levels. In addition, such as the categorical variable employment status, include NA values for 3.2% of the data. Since the variable only consist of two factors, either "Salaried" or "Self Employed". Instead of removing these observations, one new variable called "unknown" is applied for all the NA values. Lastly, one of the categorical variable, "Risk Grade Description", containing 20 classes where 13 of those represents a character corresponding to the credit rating for each individual. The remaining 7 are different reasons for why not the rating is found. Instead of creating one dummy variable for each, a score for each class in an increaseing order is generated, meaning the best creditscore gets the value 13, second-best 12,etc. As for the NA values it is set -1.

In addition to the NAs of the employment status, the dataset contains a lot of first-time borrowers. The youngest borrower is observed 18 years-old at the time of disbursement, where the oldest is 69 years old. Due to a large amount of first-time borrowers applying for a loan, there exist many zero-values in the variables of bureau history. In fact, 119 127 observations have zero credit history, which is above 51% out of the total borrowers. Because the large number of missing values, one new binary column is created, representing 1 if the borrower has loss of information, and 0 otherwise.

2.2.2 Feature engineering

In this thesis Feature Engineering is done in the sense of seeking more information out the original dataset. This process involves transforming parts of the data into clear and legible

data and facilitate the models to capture patterns within the data (Chollet, 2018). This process can potentially improve the performance of any machine learning models.

The dataset contains Primary and Secondary account information of the applicants. Primary accounts are those which the customer has taken for his personal use. Secondary accounts are those which the customer act as a co-applicant or gaurantor. After investigating the significance of the primary and secondary account information (See Appendix 2 for full test), the secondary variables are found to be insignificant. However, since financial institutions cannot afford to drop any important information a combination of the primary and secondary account informations transformed into total account informations is processed. In addition, ratios from primary and secondary data are calculated in an attempt to find patterns. Table XX below summarize the processed transformations and calculations carried out in this thesis.

Table 2.2: Feature engineering of Primary and Secondary Features

New Variable	**Transformation**
Total no. accounts	Number of Primary Accounts + Number of Secondary Accounts
Total INACTIVE ACCTS	Number of Inactive Primary Acounts + Number of Inactive Secondary Acounts
Total overdue accts	Primary Overdue Accounts + Secondary Overdue Accounts
Total Current Balance	Primary Curren Balanse + Secondary Current Balanse
Total disbursed amount	Total Primary Disbursed Amount + Total Secondary Disbursed Amount
Total Sanctioned amount	Total Primary Sanctioned Amount +Total Secondary Sanctioned Amount
Total Installment	Total Primary Instalments Amount + Total Secondary Instalments Amount
Balance disburse ratio	Total disbursed amount / Total Current Balance
Primary Tenure	Primary Disbursed Amount / Primary Installment Amount
Secondary Tenure	Secondary Disbursed Amount / Secondary Instament amount
Disb. to sact. Ratio	Total Disbursed Amount / Total Sanctioned Amount
Act. to Inact. Ratio	Total Number of Active Acounts / Total Number of Inactive Accounts
Missing	If Credit history is (=0), missing = 1, otherwise 0

Most of the variables above are continuous and self-explained by the mathematical transformation. One of the variables are binary, Missing, which is set to 1 if the new customers

applying for a loan has a lack of credit history, and otherwise 0, which occure for mostly new customers which are younger than 24 years old.

2.2.3 Feature Normalization and Principal Component

Most of the categorical variables are now scaled into 0 and 1 values, meaning that the variables are on a similar scale. As for the numerical features, there exist several outliers and a wide variation in range. In addition, the data reveal some degree of multicollinearity which, could cause a problem for some of the ML models. Principal Component Analysis (PCA) is a process where the principal components are computed and subsequently used in explanatory variables in situations with multicollinearity(James, Witten, Hastie, & Tibshirani, 2013). PCA includes normalization of the features handling the ranges of individual features. Since those are used to measure the values of the features, standarization is advisable since the large ranges will dominate over those with lower ranges (James et al., 2013). processing the features space with a mean of zero and standard deviation of one are done in an attempt to normalize the data. The calculations can be written as:

$$Standardized\ value\ of\ x_i = \frac{x_i - mean\ of\ x}{std\ dev\ of\ x}$$

According to James et al (2013), applying Principal Components scores in a statistical classification technique, is a great process to reduce the noisiness of the data because the signals in the data set is concentrated. PCA is often applied to reduce a high dimensional dataset, in a way to simplify the models and explain most of the variance of the data. Reducing the number of components or features could cost some accuracy, beacuse it reduces the proportion of the explained variance in the data. On the other hand reducing number of componets could benefit the complexity of tha data and while reducing the computational fitting process of the ML techniques(James et al., 2013). Several empirical studies have proven that normalization of numerical features have been significantly successful, Specially for the Neural Network method. Standarization have shown to be significantly successful particularly for the Neural Network method[@]. The calculation behind the Principal Components can

be described in the following formulas:

$$PC1 = w_{1,1}(Feature\ _1) + w_{2,1}(Feature\ _2) + ... + w_{n,1}(Feature\ _n)$$

$$PC2 = w_{1,2}(Feature\ _1) + w_{2,2}(Feature\ _2) + ... + w_{n,2}(Feature\ _n)$$

$$PCp = w_{1,p}(Feature\ _1) + w_{2,p}(Feature\ _2) + ... + w_{n,p}(Feature\ _n)$$

The optimal number of principal components is determined by looking at the cumulative explained variance ratio as a function of the number of components(James et al., 2013).

2.3 Splitting of data

One of the main goals of Machine Learning is to understand the general structure of the data, so that one can generate predictions about unknown features in the future. If one train and test Machine learning techniques on the same dataset, it should not be of any surprise that the results will be will be very good(Prado, 2018). A big challenge when modelling, while trying to predict the future, is to fit a model that accurately predicts applicants in the future and not over-fitting a model(Hyndman & Athanasopoulos, 2018). Overfitting a model to data is just as bad as failing to identify a systematic pattern in the data(Hyndman & Athanasopoulos, 2018).

To prevent losing the predictive power and "false" results of the models, the data is split into three datasets; training, validation, and test set, where every observation from the pre-processed data belong to only one of them. The test set is often described as the "hold-out-sample", because this data is "held out" of the data used for fitting the model, while the training set often is defined as the "in-sample data" (Hyndman & Athanasopoulos, 2018). In this thesis, "training", "validation," and "test set" is used. Splitting the data prevent leakage of information from one set to another (Prado, 2018).The training set is used to fit the model and estimate the parameters of the ML methods, and the validation set is used to tune and evaluate the accuracy of the models. The model should not overfit, however, because of the "two" testing sets, one can inspect if overfitting is present.

A common practice in machine learning is to split the data in Train, Validation and Test sets. However, uneven distribution across the data sets could lead to worse performance compared to evenly distributed sets. Thus, a randomized split of the data is set to keep the distributions stationary. This means that each set has a more or less identical portion of defaulted applicants. In addition to the evenly distributed data sets, two conditions is important to consider when a split is applied(MLCC, 2020). First, the amount of observation in each set is large enough to yield statistically meaningful results. Secondly, the validation and test set must include identical characteristics to the training set (MLCC, 2020). The splits aim to avoid overfitting while keeping enough data to generalizing the models to new unseen data. The following table represents the split and distribution of the three data sets.

Table 2.3: Split of Preprocessed Data

	Default	Non_Default	Total_Observations
Train set	21.91%	78.09%	139892
Validation set	21.44%	78.56%	46631
Test set	21.37%	78.63%	46631

The final split of the pre-processed data follows a 60/20/20 separation with a randomized generated The distribution of the response function

3 Introduction to Models

In this chapter, the estimations of the models performance will be introduced. Secondly, all models applied in this thesis will be presented in a theoretical point of view. starting with the Logistic Regression as the baseline model, moving over to the tree-based methods, support vector machines and Neural Networks. Lastly, a description of how the models have been developed will be described.

3.1 Scoring models

In classification models, there exist many different methods for evaluating a models performance, where each method offers various perspectives on the model performance (James et al. (2013)). Because our models produce probabilities for default, there is a need to apply a threshold value to separate defaulted and Non-defaulted customers. The evaluation process is initiated when a chosen threshold value is set. From the predicted classification, a confusion matrix is applied in this thesis to measure and evaluate models' performances. This matrix provides insights into the models by showing the correct and incorrect predictions of each class. The following table describes the performance of a classification model from the predictions in a two by two confusion matrix (James, 2017).

Table 3.4: Confusion Matrix

.	Actual Non-Default	Actual Default
Predicted Negative	TN	FN
Predicted Positive	FP	TP

From the confusion matrix the observed and predicted values are compared to see how well the model performs. When a model predicts a borrower as default while actual class is defaulted, then it is considered as a True Positive (TP), in other words a correctly classified default. Once the actual value is non-default, and the prediction is default, it is considered a False Positive (FP). Conversely, a True Negative is when both the predicted value and

the observed value are both non-default. Lastly, False Negative is when the predicted value is non-default, but the actual value is default. The desired outcome of the prediction is to predict alike the actual values, which means we want as many TN and TP as possible. In this thesis, a count from the Confusion matrix is applied to calculate the Accuracy, True Positive Rate (TPR), False Positive Rate (FPR).

Precision and Recall provide new insights into how well the model can predict the defaulted values. Precision assesses the number of correct predictions in the respective default class compared to the total number of predictions. Precision is the same as what someone call True Positive Rate or sensitivity. Finally, Recall assesses TP in relation to TP and FN. Recall considers the correct classifications in relation to the total number of actual classified defaults class.

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

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

The F1-scores combines Precision and Recall, which is a good measure when dealing with uneven class distribution. The F1 takes into account both False Positive and False Negative. The F1 calculations is written as $F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$. This measure is an weighted average of how well the model is when predicting defaults and how good the model is correctly predicting defaults. The scores ranges from zero to one, where 1 is the best possible outcome and 0 is the worst.

The Confusion matrix is a convenient way of visualizing how well a model is correctly classifying the actual observations.However, using a confusion matrix requires knowledge setting the threshold value for the probabilities of default(James et al., 2013). One of the most commonly used methods for evaluating Machine learning models are the Average under the Curve (AUC) (James, 2017). The AUC derives from the Receiver Operator Characteristics (ROC). The ROC curve using the True positive rate (TPR) and False Positive Rate (FPR), which is calculated as $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FN}{FN+FN}$. The TPR function is the percentage of correctly classified as default against the total number of actual defaults, while the FPR is the percentage of predicted default against the total actual non-defaulted

observations. The AUC is appropriate to determine the differance of models performance when ajusting the threshold values. The AUC ROC measurement varies the threshold values, and expresses how a model is capable of distinguishing the two classes as a function of the threshold value(Narkhede, 2018). The higher the AUC is the better the model predicts TNs and TPs.

3.2 Logistic Regression

Logistic Regression (LR) is the most widely used machine learning technique in classifications. The LR model is a linear regression model, where the independent variable is a non-linear function of the probability of the response variable(James et al., 2013). This method is a specific type of a generalized linear model, as it transforms the output using the logistic sigmoid function to which is based on the concept of probability(James et al., 2013). The scale of the coefficient are within terms of log-odds meaning we calculate the probability of true or false values. To give an intuition of the logistic regression, one variable regression model in introduced, and the Log-odds can be written as:

$$log(\frac{p(Defaulted)}{1-p(defaulted)}) = \beta_0 + \beta_1 x_1$$

From the function above, x_1 represent features from the data. A change in x_1 by one unit change the log-odds of true(default) by β_1, or equivalent it multiplies the odds by e^{β_1} (James et al., 2013). The logistic regression estimates the coefficients using a likelihood function. An example of the likelihood function can be written as:

$$\ell(\beta_0 + \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=1} (1 - p(x_{i'}))$$

Maximizing that likelihood function estimates the parameters, meaning we are identifying the coefficient for the variables. Ideally, feeding these numbers into the logistic function formula, gives a number close to one for all individuals who defaulted, and a number close to zero for all individuals that have not. The ML method will be used to fit several of the non-linear models that we are going to examine throughout this thesis (James et al., 2013).

3.3 Support Vector Machines

Support Vector machines (SVM) is a classification method, developed in the computer science community in the 1990s and has increased in prominence ever since then (James et al., 2013). It is a further extension of the support vector classifier which produces non-linear decision boundaries built on kernels. Because of our high dimensional data, the support vector classifier is a hyperplane classifying observations given its feature values. The hyperplane allows for misclassification when fitting the model when observations are not clearly separable which appear in some of the variables in our data. The details of computing the support vector classifier is somewhat complicated and technical(Gandhi, 2018). The objective of the SVM algorithm is to maximize the separation margin and minimizing the classification error. According to (Ghaddar & Naoum-Sawaya, 2017), A classical SVM formulation is to train the classifier function using pre-labeled data. Each observation in the training data has their p number of features, and a corresponding label $y_i \epsilon \{-1, 1\}$.

Advantages of Support Vector Machines is that it works well in several different settings, and does not necessarily require much of computational power. In addition, the proposed feature selection and SVM classification is computationally tractable and easy to incorporate, which is important when dealing with a large number of variables(Ghaddar & Naoum-Sawaya, 2017).

3.4 Tree-Based Methods

This method has been around in many fields for a long time without formal statistical or mathematical underpinning, but during the last decade decision trees have been introduced as one of the baseline machine learning algorithms for regression and categorical data(Olbricht, 2012). Tree-based methods, also known as decision trees are techniques to segment predictors space into simple regions. It is based on if-else questions of individual features, which are called nodes. Decision trees separating the observations where it first divides the predictor space, the set of possible values for $x_1, x_2, \ldots x_p$, into J distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$. Next we make a prediction based on every observation falling into region R_j (James et al., 2013). Selecting the optimal model is decided by minimizing the classification

error rate with a cross validation test in the traing data. Each feature segmentation is set through different splitting rules, where the objective is to minimize quantities such as the gini index, classification error rate or cross-entropy(Hastie, Tibshirani, & Friedman, 2009; James et al., 2013).

Decision trees provide a clear and useful approach to data analysis, but are not necessarily compatible with the more effective techniques of machine learning(James, 2017). According to Breiman et al (1993) trees can be optimized when using a learning set of the observation to prune the saturated tree and select among the so obtained sequence of nested trees fitted to the appropriate training set (learning set)(Breiman, Friedman, Stone, & Olshen, 1993). This process helps to maintain a simple tree while guaranteeing robustness. In addition, bootstrap, random forests, and boosting use trees as building blocks to construct more powerful prediction models, which will be described in detail in the following section.

3.4.1 Random Forest with Bootstrap aggregation

Bootstrap aggregation, or *bagging*, is a general-purpose technique for reducing the uncertainty of a statistical learning method which is frequently used in decision trees (James et al., 2013). The main idea behind Bagging is to take repeated samples from the original training data set, and generate different bootstrapped training data set. Then by training our method based on the the bootstrapped training set we can record the predictions, and choose the most commonly occurring class as the overall prediction(James et al., 2013).

As a method of decorrelating the leaves, random forests provide an improvement over simple bagged trees when very strong predictors may produce similar bagged trees that not generate a reduction in variance(James et al., 2013). Hence the Random Forest (RF) method consider only a subset of the predictors, henceforth the word “random”. This allows other strong predictors to classify, generating a reduction in the uncertainty of the predictions. Lastly, the RF method is beneficial when dealing with a large number of correlated features to reduce the variance (James et al., 2013).

3.4.2 Gradient boosting

Friedman (1999) introduced a further development of the Adaptive Boosting or short for AdaBoost algorithm. This algorithm was called Gradient Boosting Machines (GBM), which in recent years are called Gradient Boosting. GBM is created so that observations that are slightly better than a random choice (weak learners) are trained to augment each other and produce superior results(Friedman, 1999). The major difference between the two methods is how they identify the deficiencies of the weak learners. The GBM algorithm identifies the lacks of the weak learners by using the gradients in the loss function, while the AdaBoost using high weight data points (Singh, 2018). As the loss function is an estimate signifying how well the model is predicting credit defaults.

The GBM model differs from the RF model where the trees are learning based on prior grown trees sequentially and base the new tree on their performance converting weak learners into strong ones(Chen & Guestrin, 2016). GBM has outperformed other machine learning algorithms within a number of data challenges, and in addition, recent work has described its potential within the medical field (Klug et al., 2020; Singh, 2018).

The Gradient boosting algorithm applied in this thesis involves three parameters. (1) A loss function to be optimized (2) A weak learner to make prediction. (3) An additive model to add weak learners to minimize the loss function. The process of the Gradient boosting algorithm, which is applied in this thesis, initiates by training a decision tree where each observation is given equal weights to classify. Then assessing the tree, detecting which observations that are hard to classify (weak learners) and increase their weight, contradictory for the easier observations. Further, the second tree will then be built on these new weights, therefore learning by the previously grown tree(s). As the sequence of trees grows in a specified number of times, the aim of the algorithm is to improve the prediction upon the previously built tree.

3.5 Deep Learning

Deep learning methods are somewhat more advanced compared to Machine Learning (ML). However, ML methods could be portrayed as a super-set of deep learning, meaning it works with data as input, and parse the data, to make sense of the decisions based on what it

has learned. Whereas Deep Learning is a subset of machine learning methods, using an artificial neural network stacked layer-wise to make analytical and intelligent decisions (Singh, 2018).One of the reasons for applying Deep learning methods in this thesis is because of the amount of datain our application. ML models perform good even with a small amount of data, however, Deep Learning methods are so-called "data hungry", meaning that having more data tends to improve the performance of the model (Singh, 2018). In this thesis, Neural Network will be applied as the one deep Learning technique for predicting defaults of the car loans. This approach has not been commonly used in PD modelling, however it has found many business applications in recent years (Tkáč & Verner, 2016).

3.5.1 Neural Network approach

The Neural Network (NN) approach, as mentioned, working with layers which is set of algorithm design to recognize patterns within the data. A Neural network is illustrated to provide the reader with an overview of the process of the NN approach. As seen below, this network consists of one input layer, p number of hidden layer, and one output layer. The input layer is the given variable within a given dataset. The hidden layer is a way to transform the data to achieve a good approximation of the predicted values.

In a neural network, we are able to tune the number of hidden layers, the activation function for different layers, the number of hidden units for each layer, and lastly, the learning rate. The number of hidden layers represent the depth, while the number of hidden units (vertical nodes) represent the width of the network (Goodfellow, Bengio, & Courville, 2016). In our case, the 39 variables in the dataset is corresponds to the number of input layers. Each of the input layers are connected to every hidden units within the first layer, which cause a chain towards all layers. These connections are weighted (not illustrated in the figure) meaning the model emphasis each layer and units to reflect data patters to which prediction accuracy could improve (Mantovani et al., 2018). Weights are learned by attempting a lot of different numbers to minimize the error. The weights change the formula by moving poor-performing weights back and well-performing weights to a higher degree. Lastly, the output layer collects the features from the last hidden layer and executes the last conversion, typically exercising

a "Softmax activation" function that converts the features into a probability distribution (Chollet, 2018).

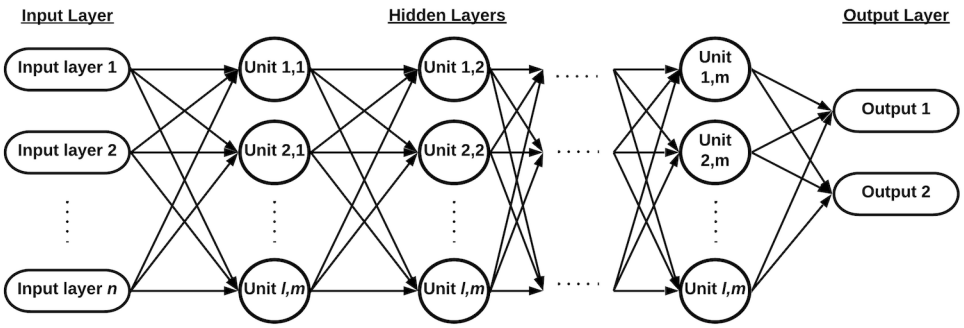

Figure 3.1: Neural Network with Multi Hidden Layers and Units

3.6 Technical aspect

In this thesis, the process of building models and processing the data is done using R via R Studio. As for the machine learning techniques, the R-packages; stats, randomForest, gbm, e1071 and neuralnet is applied to build the logistic regression, Random Forest, Gradient boosting, Support Vector Machine and Neural Network. Lastly, a MacBook Pro with a 2.7 Ghz intel core i5 processor and 8GB memory disk, is the operating computer used to run the analysis.

3.7 Development of the models (Hyperparameter tuning)

As we mentioned above, many of the machine learning techniques include several configurations known as hyperparameters. These parameters control the operations of the model and how it executes it tasks, while impacting the performance (Kantardzic, 2019). In other words, ways of how the algorithms are restricted to be built and learned. Finding the optimal parameter, can be challenging and time-consuming, especially when dealing with a large amount of data and an advanced model. The goal is to define strong parameters so unknown data can be predicted correctly by the classifier(Kantardzic, 2019). For the majority of the models used, we proceed with a grid search, iterating across values of the parameters. Based

on the accuracy of the model, the hyperparameter values will be chosen. However, for certain algorithms applied in this thesis, there exist many more parameters to tune, but based on literature, we chose the set of parameters that gives the best impact of the modelperformance. The table XXX presents the grid search for all parameters in each model.

Similar to other Machine Learning algorithms, decision trees include hyperparameters to be set to affect the predictive performance of the induced models. Optimization methods are used to dynamically check for an optimal range of hyperparameter settings(Mantovani et al., 2018). However, due to the fast run time of decision trees, a grid search is used to find the optimal parameters of the model. A grid search is a try and fail approach, which manually needs to be set up with intervals for each parameter (Mantovani et al., 2018).The best combination that performs with the highest accuracy, will then be chosen and tested on the test set. This procedure has been used for several studies, which has shown great records(Feurer et al., 2015; Thornton, Hutter, Hoos, & Leyton-Brown, 2013). For the Random Forest model, the main hyperparameters is chosen; ntree, mtry, nodesize and max_debt. The ntree parameter is the number of trees to grow. Mtry is the parameter chosing the number of variables randomly sampled as candidates at each split. The nodesize argument interacts with the size of the trees nodes, and characterizes the minimum terminal node size. Setting a high value for this parameter causes smaller trees to grow, compared to the default value of 1. Lastly, the max_debt parameter specifies the maximum debt of the trees to be allowed. As for the last tree-based algorithm, Gradient Boosting, there exist many parameters to tune. The parameters to tune in the application of this thesis is; number of trees, shrinkage, and depth. The shrinkage parameter is different from the other tree-based method, as it shrinks the contribution of each successive decision tree in the ensemble. The grid-search for both the Random Forest and Gradient Boosting is chosen based on previous study and reasonable intention. In table XXX is a summary of all the models' hyperparameters and the respective grid search.

The Support Vector machine contains many more parameters to tune in contrast to other ML techniques. Despite all the potential parameters to tune, the cost and Kernel parameters are the most vital(Kantardzic, 2019). The Cost parameter decides the degree of

misclassification, allowing the model to partake. A too-small Cost parameter could result in underfitting the learning. If we chose a too large Cost value, the classification margin would be too thin, resulting in overfitting the learning(Kantardzic, 2019). The same is for the kernel parameter. A small value will lead to a linear kernel involving no significant transformation of the data, while a very Large value could generate too complex nonlinear solutions (Kantardzic, 2019). The last hyperparameter for the SVM predictions is the gamma parameter. The gamma parameter determines how frequent the dissipation of every individual support vectors, increasing the Gamma value decreases the effect of dissipation(Chapelle, Vapnik, Bousquet, & Mukherjee, 2002).

The Neural network algorithm can be more complex and more time consuming than the other models introduced in this thesis. For that reason, a more manually and individual parameter tuning is used. Professor Andrew Ng suggested a guideline tuning parameters for the neural networks algorithms, explicitly tuning the three most impactful parameters; learning rate, number of hidden layers and number of hidden units[Liu (2019);Goodfellow et al. (2016). Following his guide of tuning the parameters start by the using grid search for the Learning rate. This value has a small positive value, often in the range of 0 and 1(Brownlee, 2019). The learning rate regulates the speed at which the model adapts to the problem. Low rates require more training because of the small change in weights each update, while larger rates result in rapid changes (Brownlee, 2019). The default value of the learning rate is set to be 0.01, which has shown to be a decent rate (Goodfellow et al., 2016). For the number of hidden layers, the increase in layer generate more complex algorithms. Jeff Heaton (2017) states that the use of more than two layers are rare, which only benefit cases with complex dataset with time series or computer vision. Lastly, the number of hidden units is very important in deciding the overall NN architecture, which needs to be set to completely to adequately detect the signals and patterns in the data(Heaton, 2017). Setting a small number of hidden units, could result in underfitting, while setting a too large amount can lead to overfitting as the hidden units needs enough information to be trained(Heaton, 2017). Keeping Heaton´s rule-of-thumb in mind, the hidden units should be between the number of independent variables of the dependent variable.

The following table present the grid search of

Table 3.5: Hyperparameter and the grid search of RF,GBM,SVM and NN

Model	**Hyperparameter**	**Grid Search**
RF	Ntree	[100,500,1000,5000]
RF	Nodesize	[2,5,10,15,20]
RF	mtry	[0,2,5,9]
RF	max depth	[2,5,7,9]
GBM	ntree	[100,500,1000,5000]
GBM	shrinkage	[0.01,0.1,10
GBM	max depth	[2,5,7,9]
SVM	Cost	[0.01,0.1,10]
SVM	Gamma	[0.01,0.1,10
SVM	Kernel	Radial, RBG
NN	Learning Rate	[0.001,0.01,0.1]
NN	Hidden Layers	[1,2,3]
NN	Hidden Units	[2,8,14,20,26,32,38]

4 Analysis

4.1 A broad look at the performance

The overall performance for all models are illustrated in the table below. The Logistic Regression, also refered to the baseline model in this thesis, obtain the second-best results with an accuracy of 78,50% in the test set. As for the decision-tree based methods, the Gradient Boosting machine perform best out of the two methods. The performance table illustrates the GBM's accuracy which is 75.59% while the Random Forest followes closely with an accuracy of 74.89%. Compared to the Logistic regression, the differance is marginal. The Support Vector Machines performs best out of all the models which is observed at 78.61% accurate. As for the Neural Network the performance of overall accuracy is observed at 78.38%. The differance is marginally, so we look closer at the AUC values.

Table 4.6: Performance of Accuracy and AUC in Test Set

Method	**Accuracy**	**AUC**
LR	0.7850	0.6325
RF	0.7489	0.5969
GBM	0.7559	0.6086
SVM	0.7861	0.5015
NN	0.7838	0.6349

As described above, the SVM is the most accurate model. However, the AUC measurements indicate a better performance of the Neural Networks, compared to the Support Vector Machines. As mentioned in chapter 3, the AUC estimations express the quality of a classifier, in terms of how well a model perform against "random picking", which is a good measure to compare models. As we can see from the figure below, all models are visualized in therms of AUC values. In the figure, the diagonal line from the lower left corner towards the upper right corner indicates a random choice. The Neural Networks, orange line, observed an AUC of 0.6349, while for the Logistic regression 0.6325. The AUC measurement indicates that the NN

are the best classifier, meaning that the model is slightly better to classify defaulted applicants compared to the Logistic Regression. As for the Support Vector Machines, the observed AUC value is 0.5015, which is the worst performance out of all five models. The results of the SVM algorithm is somewhat surprisingly low, which indicates that “random guessing” is almost equally good as the SVM. Lastly, For the two tree-based methods, Random Forest and Gradient Boosting, the performance for them both are better than a random choice, however, slightly less AUC values of 0.5969 and 0.6086, compared to the NN and LR. Tree-based methods have their status to perform very well in many circumstances, specially the random Forest which is spoken to among the most powerful machine learning techniques (James, 2017). as the NN perform good, could indicate a Non-linear pattern in the data, however, the LR model which, in theory" should not be able to perform well in major non-linear problem, it seems to that the LR are able to capture most of it.

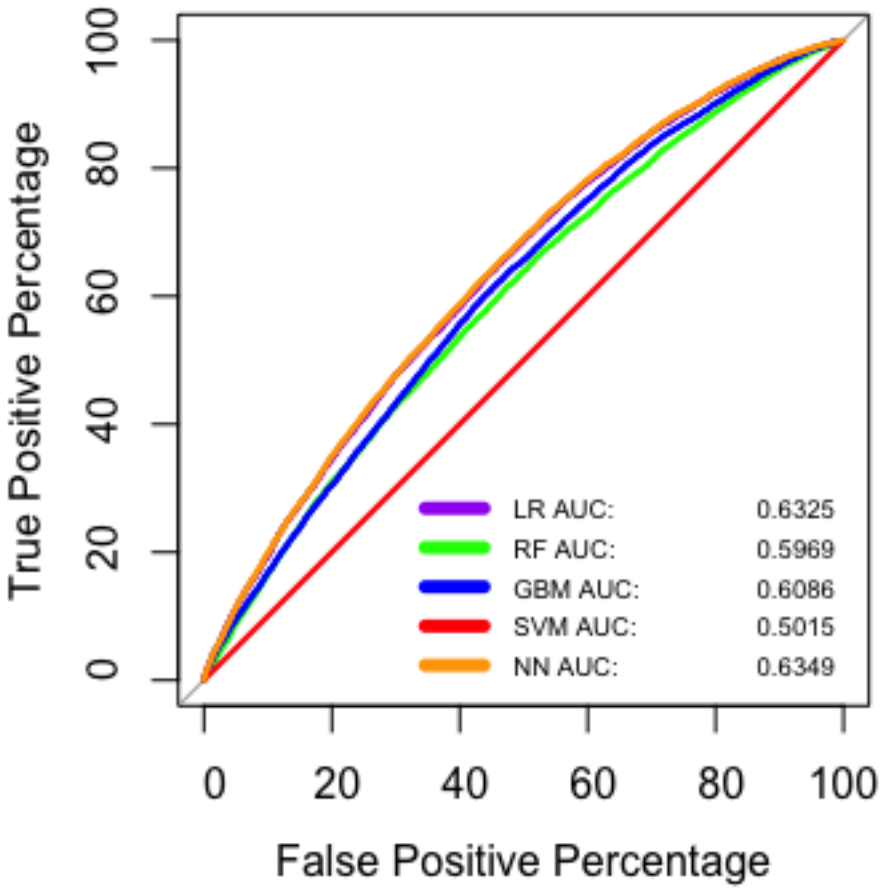

Figure 4.2: AUC Performance of LR, RF, GBM, SVM and NN from the Test set

4.2 Further investigation of all models

As laid out in section XX, the models performance in the matter of Recall, Precision and F1 scores is needed to analys the models performance. As described, Recall estimate the correctly classified default to the total of actual defaulted applicants, or in other words the proportion of actual defaulted applicants the model captures. As for Precision, it examine the portion of correctly classified defaults to all classified defaults. Lastly, F1 Scores estimate the weighted average of the performance by both Precision and Recall. As one can observe from the function in section XX $F_1 - Score$ will only be high if both precision and Recall are

high. The table below provides an assessment of the three performance metrices in the test set.

Table 4.7: Recall, Precision and F1 scores from the Test set

Method	Precision	Recall	F1Scores
LR	0.4494	0.0272	0.0513
RF	0.3109	0.1439	0.1967
GBM	0.3233	0.1299	0.1854
SVM	0.4545	0.0045	0.0089
NN	0.4280	0.0340	0.0630

As displayed in the table XX, all of the models struggle to capture a well proportion of the actual defaulted applicants- in other words, recall values are low. As for the Precision, the models are better to capture the defaulted applicants. The *Support Vector Machines* is worst in terms of Recall, as observed from the table above, the model measures 0.45% of the total defaulted applicants. The Precision measurement indicates that the model hits 45.45% of the predicted defaulted applicants is correctly classified, while the rest is non-defaulted customers. However, the F1 scores are supprisigly low at 0.0089. As for the Tree-based models, the *Gradient boosting machines* and the *Random Forest* seems to perform better in terms of F1 scores. The Precision from the GBM is observed at 32.33%, and Recall are less accurate with 12.99%. As for the *Random Forest* , the model obtain the worst Precision out of all the models, however best in terms of Recall. RF´s Precision and Recall is observed at 31.09% and 14.39%. However, in terms of f1-score, the RF models perform best, meaning that the model are better in capturing the default on average. As for the baseline model, the Logistic Regression´s Precision are the second most accurate model with 44.94% proportion of correctly predicted defaulted applicants. However the Recall of the LR model is observed at 2.72%, meaning that the model captures only 2.72% of the all actual defaulted borrowers. Lastly, the Neural Network with the highest AUC, capture 3.4% in Recall and 42.80% in Precision.

As we can see from the measurements, all models struggle to capture a major proportion of the actual defaulted applicants, however the LR, SVM and NN are the most precise in terms of correctly predicted defaults (Precision). Looking at the F1 Score, the tree-based models predicts quite good, with the highest weighted average of precision and recall. Lastly to mention, that the Neural Network and the Logistic Regression generate the best AUC. The following figure visualize the confusion matrix of the two best models in terms AUC, F1-Score, respectively, while including the Logistic regression.

Table 4.8: Confusion Matrix of The Three Best Models

	LR		**RF**		**NN**	
	Non-Default	**Default**	**Non-Default**	**Default**	**Non-Default**	**Default**
Predicted Non-Default	36333	9695	33486	8532	36212	9627
Predicted Default	332	271	3179	1434	453	339

4.3 Influence in the credit industry

A relevant question within the modelling procedure would be if any model influence the decision-making process for the creditors. As we can see from the model´s performance, in terms of AUC the Neural Network perform better that the other. However, it is worth to mention the drawbacks of the Neural Network. As it is difficult to interpret the model, a creditor would struggle to describe the reason why not an applicant would not be granted a loan. This problem occur in several ML algorithms, in addition to the principal component analysis which generate components that transform of the features as described in section XX. The complexity to back-transform the original numbers are not impossible, however may require some knowledge and computations from the preprocessing process. In addition to the Neural Network, the three-based methods gets more complex as the number of threes expands. However, the Logistic Regression are performing quite well, in terms of AUC, and easier to interpret. However, du to its limitations, the Neural Network, Random Forest and Logistic regression perform better then random picking, thus are able to contribute in the decision-making process.

5 Discussion

The analysis of the results indicated that machine learning are valuable when classifying the credit risk in the automotive industry. This study provides insights in some of the machine learning models in terms of predicting defaults. However some limitations are present in the current solution, and some interesting ideas for future research. First, presenting the validity of the models performance, secondly some limitations of the models and data, lastly offer some interesting ideas for future research.

5.1 Validity of the models

In terms of Validity of the models, as described in section XX, the data set were split into three unique data sets with more or less equally portions of defaulted applicants. Other litterature presents different approaches tuning the hyperparameters of the models. For example Cross Validation is introduced by James (2013), which could be argumented towards better results. However, an Cross Validation approach would train the model in sequences of the whole data set. Such an approach would be time consuming, due to the complexity of models such as Neural Network and Support Vector Machine, to mention two. However, splitting the dataset into three unique data set prevents leakage of any information and avoid overfitting. The tables below presentes each models perfromance score in respective Validation and test set. The split was important in the case of testing the model on unseen data and to compare the models to each other. the two following tables presents the performance of all the model in the Validation and Test set.

Table 5.9: Validation Set Performance of all models

Method	Accuracy	AUC	Precision	Recall	F1Scores
LR	0.7837	0.6263	0.4299	0.0273	0.0514
RF	0.7482	0.5951	0.3128	0.1457	0.1988
GBM	0.7815	0.6313	0.4204	0.0510	0.0910
SVM	0.7852	0.5014	0.4128	0.0045	0.0089
NN	0.7834	0.6318	0.4356	0.0352	0.0652

Table 5.10: Test Set Performance of all models

Method	Accuracy	AUC	Precision	Recall	F1Scores
LR	0.7850	0.6325	0.4494	0.0272	0.0513
RF	0.7489	0.5969	0.3109	0.1439	0.1967
GBM	0.7559	0.6086	0.3233	0.1299	0.1854
SVM	0.7861	0.5015	0.4545	0.0045	0.0089
NN	0.7838	0.6349	0.4280	0.0340	0.0630

As one can observe, the performance of the models seems to differ by only a few decimal points, when looking at the model´s perfromance in the two data sets. The only outlier in differance of the validation and test set occur in the GBM. In terms of AUC, precision and recall, the GBM perform better in the Validation set, than in the test set. These results may indicate that the model are tuned too well to the validation set, and may not be as good to predict on unseen data. Despite that this may indicate that the model does not generalize, the other models seems to be better.

5.2 Limitations of models and data

The data was collected from Kaggle, which is an open database source, which may question the reliability of real world data. In addition, the data set contained a big proportion of NA values in some the credit-history features, which may affect the quality of the data. However, such a problem may occur in the future for creditors, for example when young applicants apply for a loan with zero credit history.

As for the hyperparameter tuning, most of the models were tuned with the grid search. However, for the Neural Network and Support Vector Machines, the tuning process was very time consuming. As for the Neural Network, when exceeding the numbers of hidden layers for more than 1, the fitting process took very long time to run. The potential of the NN and SVM could have been improve even further if the grid search would have been finished. In addition, each models were only tuning mostly three parameters. the existance of several paramters could have improved the models. in addition one could argue about the grid search technique as there exist other optimization techniques to search for the optimal parameter values.Lastly, the grid seach are initiated with steps, meaning that a more narrow search could have improve the models even further.

5.3 Further Reseach

As we have seen from the performance of the models, the improvement from the dataset of terms of accuracy, were not very impressive as the all ready accuracy were almost 80%. However, the Neural Network is greater in detecting the default applicants. As the dataset containing already granted loans from banks or other financial institutios, its clear that banks are allready stuggeling to detect these defaulted customers. To detect these customers, one may think of more information the borrowers would potentially improve the predictions. Important information such as Income status are not included in the data set, and could potentially improve the prediction, which could be of interest in future reseach. Other researchers have pointed out that the one should shift the focus from PD models to other modeling problems in the credit industry; including data quality, scorecard recalibration, variable selection, and Loss Given Default modelling (Lessmann et al., 2015). Improving the

data quality, while collecting more information of the customers, could benefit the creditors profitability, due to more precise data.

6 Conclusion

The purpose of this master theis was to investigate the current machine learning methods, which has proven to be good techniques predicting default, and examine their response in the automotive credit industry. Throughout this study, machine learning algorithms such as Logistic regression, Random Forest, Gradien Boosting Machines and Neural Network have been examined finding the Neural Network to perform best, with an AUC of 0.6349. As for the baseline model, the Logistic regression follows the Neural Network, with an AUC of 0.6325, and the second best model. However in terms of accuracy, all of the models are perform good. The results from the analysis indicated that the Logistic Regression are better than most of the other models, except for the Neural Network that perform marginally better. Due to that the Logistic Regression are easier to interpret and generate almost equally good results as the Neural Network, the Logistic regression may be applied in future default predictions within the automotive credit industry, due to its well performing model as it captures the potential non-linear patterns of the data. However, as the data quality and computers may be improved in the future, the Neural Network might be a better choice due to its well performance in high dimentional data.

7 References

Agarwal, S., Ambrose, B. W., & Chomsisengphet, S. (2008). Determinants of automobile loan default and prepayment. *Economic Perspectives - Federal Reserve Bank of Chicago*.

Agrawal, A., Agrawal, M., & Raizada, D. A. (2014). Predicting defaults in commercial vehicle loans using logistic regression: Case of an indian nbfc. *International Journal of Research in Commerce and Management*, *5*, 22–28.

Aleum Kim, S.-B. C. (2019). An ensemble semi-supervised learning method for predicting defaults in social lending. *Engineering Applications of Artificial Intelligence*, *81*, 193–199.

Altman, E. I., & Saunders, A. (1998). Credit risk measurement: Developments over the last 20 years. *Journal of Banking and Finance*, *21(11-12)*, 1728–1742.

Bekhet, H. A., & Eletter, S. F. K. (2014). Credit risk assessment model for jordanian commercial banks: Neural scoring approach. *Review of Development Finance*, *4*(1).

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1993). *Classification and regression trees*. New York: Chapman; Hall.

Brownlee, J. (2019). Understand the impact of learning rate on neural network performance. *Deep Learning Performance*, *1*.

CFI. (2020). Non-performing loan (npl): A loan in which the borrower is in default. Corporate Finance Institution. Retrieved from https://corporatefinanceinstitute.com/resources/knowledge/finance/non-performing-loan-npl/

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, *46*(1), 131–159. doi:10.1023/A:1012450327387

Chen, M.-C., & Huang, S.-H. (2003). Credit scoring and rejected instances reassigning throughevolutionary computation techniques. *Expert Systems with Applications*, *24*.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *CoRR*, *13-17*.

CHEN, Y. (2017). Research on the credit risk assessment of chinese online peer-to-peer lending borrower on logistic regression model. *DEStech Transactions on Environment, Energy and Earth Science*.

Chollet, F. (2018). *Deep learning with r*. Shelter Island, N.Y: Manning Publications Co.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, *183*, 447–1465.

Experian. (2020). Auto loan debt sets records highs. Retrieved from https://www.experian.com/blogs/ask-experian/research/auto-loan-debt-study/

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems 28*

(pp. 2962–2970). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

Friedman, J. (1999). *Greedy function approximation: A gradient boosting machine*. Retrieved from https://statweb.stanford.edu/~jhf/ftp/trebst.pdf

Gandhi, R. (2018). Support vector machine — introduction to machine learning algorithms. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Ge, R., Feng, J., Gu, B., & Zhang, P. (2017). Predicting and deterring default with social media information in peer-to-peer lending. *Journal of Management Information Systems*, *34*(2), 401–424. doi:10.1080/07421222.2017.1334472

Ghaddar, B., & Naoum-Sawaya, J. (2017). High dimensional data classification and feature selection using support vector machines.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning (adaptive computation and machine learning series)*. MIT Press.

Hao, C., Alam, M. M., & Carling, K. (2010). Review of the literature on credit risk modeling: Development of the past 10 years. *Banks and Bank Systems*, *5*(3).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Element of statistical learning: Data mining, inference and prediction*. (S. Edition, Ed.). Springer.

Heaton, J. (2017). The number of hidden layers. *Heaton Research*.

Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.com/fpp2: OTexts: Melbourne, Australia.

James, G. (2017). *Machine learning in credit risk modelling*. White Paper.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning with application in r*. (G. Casella, S. Fienberg, & I. Olkin, Eds.) (Vol. 8th). Springer.

Jiang, C., & Zhang, S. (2017). Absolute phase unwrapping for dual-camera system without embedding statistical features. *Optical Engineering*, *56*(9), 1–7. doi:10.1117/1.OE.56.9.094114

Kantardzic, M. (2019). *Data mining: Concepts, models, methods, and algorithms*. John Wiley & Sons, Incorporated.

Klug, M., Barash, Y., Bechler, S., Resheff, Y. S., Tron, T., Ironi, A., ... Klang, E. (2020). A gradient boosting machine learning model for predicting early mortality in the emergency department triage: Devising a nine-point triage score. *Journal of General Internal Medicine*, *35*(1), 220–227. Retrieved from http://search.proquest.com/docview/2311650822/

Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert System with Applications*, *40*, 5125–5131.

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136.

Li, J., Hsu, S., Chen, Z., & Chen, Y. (2016). Risks of p2p lending platforms in china: Modeling failure using a cox hazard model. *The Chinese Economy*, *49*(3), 161–172. doi:10.1080/10971475.2016.1159904

Lin, X., Li, X., & Zheng, Z. (2017). Evaluating borrower's default risk in peer-to-peer lending: Evidence from a lending platform in china. *Applied Economics*, *49*(35), 3538–3545. doi:10.1080/00036846.2016.1262526

Liu, G. (2019). Optimizing neural networks — where to start? Retrieved from https://towardsdatascience.com/optimizing-neural-networks-where-to-start-5a2ed38c8345

Mantovani, R. G., Horváth, T., Cerri, R., Junior, S. B., Vanschoren, J., & Carvalho, A. C. P. de L. F. de. (2018). An empirical study on hyperparameter tuning of decision trees.

MLCC. (2020). Machine learning crash course. Retrieved from https://developers.google.com/machine-learning/

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

Olbricht, W. (2012). Tree-based methods: A useful tool for life insurance, Eur. Actuar. J. (2012) 2:129–147. Retrieved from https://link.springer.com/content/pdf/10.1007/s13385-012-0045-5.pdf

Paul, A. (2019). Vehicle loan default prediction. Retrieved from https://www.kaggle.com/avikpaul4u/vehicle-loan-default-prediction/kernels

Prado, M. L. de. (2018). *Advances in financial machine learning.*

Singh, H. (2018). Understanding gradient boosting machines. Retrieved from https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

SRD. (2020). Number of motor vehicles registered in the united states from 1990 to 2018. Retrieved from https://www.statista.com/statistics/183505/number-of-vehicles-in-the-united-states-since-1990/#statisticContainer

Thornton, C., Hutter, F., Hoos, H., & Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. *KDD*. doi:10.1145/2487575.2487629

Tkáč, M., & Verner, R. (2016). Artificial neural networks in business: Two decades of research. *Applied Soft Computing*, (https://doi.org/10.1016/j.asoc.2015.09.040).

Appendix 1 – Description of original variables

Variable Name	Description
UniqueID	Identifier for customers
loan_default	Payment default in the first EMI on due date
disbursed_amount	Amount of Loan disbursed
asset_cost	Cost of the Asset
ltv	Loan to Value of the asset
branch_id	Branch where the loan was disbursed
supplier_id	Vehicle Dealer where the loan was disbursed
manufacturer_id	Vehicle manufacturer(Hero, Honda, TVS etc.)
Current_pincode	Current pincode of the customer
Date.of.Birth	Date of birth of the customer
Employment.Type	Employment Type of the customer (Salaried/Self Employed)
DisbursalDate	Date of disbursement
State_ID	State of disbursement
Employee_code_ID	Employee of the organization who logged the disbursement
MobileNo_Avl_Flag	if Mobile no. was shared by the customer then flagged as 1
Aadhar_flag	if aadhar was shared by the customer then flagged as 1
PAN_flag	if pan was shared by the customer then flagged as 1
VoterID_flag	if voter was shared by the customer then flagged as 1
Driving_flag	if DL was shared by the customer then flagged as 1
Passport_flag	if passport was shared by the customer then flagged as 1
PERFORM_CNS.SCORE	Bureau Score
PERFORM_CNS.SCORE.DESCRIPTION	Bureau score description
PRI.NO.OF.ACCTS	count of total loans taken by the customer at the time of disbursement
PRI.ACTIVE.ACCTS	count of active loans taken by the customer at the time of disbursement
PRI.OVERDUE.ACCTS	count of default accounts at the time of disbursement
PRI.CURRENT.BALANCE	total Principal outstanding amount of the active loans at the time of disbursement
PRI.SANCTIONED.AMOUNT	total amount that was sanctioned for all the loans at the time of disbursement
PRI.DISBURSED.AMOUNT	total amount that was disbursed for all the loans at the time of disbursement
SEC.NO.OF.ACCTS	count of total loans taken by the customer at the time of disbursement
SEC.ACTIVE.ACCTS	count of active loans taken by the customer at the time of disbursement
SEC.OVERDUE.ACCTS	count of default accounts at the time of disbursement
SEC.CURRENT.BALANCE	total Principal outstanding amount of the active loans at the time of disbursement
SEC.SANCTIONED.AMOUNT	total amount that was sanctioned for all the loans at the time of disbursement
SEC.DISBURSED.AMOUNT	total amount that was disbursed for all the loans at the time of disbursement
PRIMARY.INSTAL.AMT	EMI Amount of the primary loan
SEC.INSTAL.AMT	EMI Amount of the secondary loan
NEW.ACCTS.IN.LAST.SIX.MONTHS	New loans taken by the customer in last 6 months before the disbursment
DELINQUENT.ACCTS.IN.LAST.SIX.MONTHS	Loans defaulted in the last 6 months

AVERAGE.ACCT.AGE	Average loan tenure
CREDIT.HISTORY.LENGTH	Time since first loan
NO.OF_INQUIRIES	Enquries done by the customer for loans

Appendix 1 – T Test of original Numerical Variables

Var	p_value	STDERR	t_value	legreeoffre	avgDef1	avgDef0	LowInt	an_defaultI	method	alternative
disbursed_	0	62.15213	-39.3229	86243.48	56270.47	53826.47	-2565.82	-2322.19	Welch Two	two.sided
asset_cost	3.52E-12	94.21097	-6.95641	81988.52	76378.18	75722.81	-840.023	-470.718	Welch Two	two.sided
ltv	0	0.053433	-51.078	89749.4	76.88332	74.15409	-2.83396	-2.6245	Welch Two	two.sided
branch_id	4.07E-46	0.358525	-14.2661	78454.81	76.94057	71.82583	-5.81744	-4.41203	Welch Two	two.sided
PERFORM_	1.9E-186	1.628657	29.19454	85826.17	252.2364	299.7843	44.35574	50.74005	Welch Two	two.sided
PRI.NO.OF.	7.65E-69	0.025565	17.55184	83690.23	2.089328	2.538038	0.398602	0.498816	Welch Two	two.sided
PRI.ACTIVE	4.4E-109	0.008786	22.21809	95092.6	0.88706	1.082271	0.17799	0.212432	Welch Two	two.sided
PRI.OVERD	2.05E-75	0.002957	-18.3969	73820.9	0.199146	0.144738	-0.0602	-0.04861	Welch Two	two.sided
PRI.CURREI	3.06E-56	3959.189	15.80998	109735.4	116892.9	179487.6	54834.75	70354.66	Welch Two	two.sided
PRI.SANCTI	0.001355	20323.56	3.204158	52712.88	167519.6	232639.5	25285.53	104954.3	Welch Two	two.sided
PRI.DISBUR	0.001554	20331.49	3.16462	52731.79	167691.1	232032.6	24491.54	104191.4	Welch Two	two.sided
SEC.NO.OF.	5.16E-06	0.002797	4.558447	97679.16	0.0491	0.061848	0.007267	0.01823	Welch Two	two.sided
SEC.ACTIVE	0.002011	0.001488	3.088742	89127.96	0.024105	0.0287	0.001679	0.00751	Welch Two	two.sided
SEC.OVERD	0.506404	0.000556	0.664451	81294.3	0.006955	0.007324	-0.00072	0.001459	Welch Two	two.sided
SEC.CURRE	0.000348	638.6589	3.576519	142881.8	3639.446	5923.621	1032.416	3535.934	Welch Two	two.sided
SEC.SANCTI	0.000134	739.1741	3.819271	119844.1	5085.632	7908.738	1374.337	4271.876	Welch Two	two.sided
SEC.DISBUR	0.000174	737.1488	3.754278	119753.1	5013.272	7780.734	1322.662	4212.262	Welch Two	two.sided
PRIMARY.IN	4.9E-09	666.162	5.851207	100192	10053.74	13951.59	2592.183	5203.521	Welch Two	two.sided
SEC.INSTAL	0.350645	62.59433	0.933343	120616.2	277.5282	335.9502	-64.2619	181.1058	Welch Two	two.sided
NEW.ACCTS	1.21E-50	0.004548	14.97644	87431.97	0.328506	0.396619	0.059199	0.077027	Welch Two	two.sided
DELINQUEN	1.12E-52	0.002102	-15.2874	72608.02	0.122641	0.090505	-0.03626	-0.02802	Welch Two	two.sided