UNIVERSITY OF TWENTE

MASTER THESIS

Added value of machine learning in retail credit risk

Author:
D. Gorter

Supervisors:
B. Roorda
M. van Keulen
F. Reuter
M. Westra

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the field of

Financial Engineering and Management

May 4, 2017

"One problem in the field of statistics has been that everyone wants to be a theorist. Part of this is envy - the real sciences are based on mathematical theory. In the universities for this century, the glamor and prestige has been in mathematical models and theorems, no matter how irrelevant. "

L. Breiman

University of Twente

Abstract

BMS
IEBIS

Master of Science

Added value of machine learning in retail credit risk

by Derk GORTER

This thesis aims to pinpoint the added value of machine learning in the domain of retail credit risk, where (logistic) regression approaches are most commonly used. Credit data of the on-line peer to peer lending platform Lending Club is used to create retail credit risk models with Logistic Regression, Random Forests, Neural Networks and Support Vector Machines. A level playing field is created for the models by means of a single data transformation to keep the input of all models equal. This level playing field is achieved by using Weight of Evidence to create a scaled data set without outliers or missing values. The created retail credit risk models are evaluated in terms of modeling approach and in terms of model performance in order to find added value. The research shows that the added value of the machine learning approach over the traditional (logistic) regression approach is present. Where the machine learning algorithms can handle all variables and decide for themselves how to model the relationships between the variables, the (logistic) regression approaches need careful selection of subsets of independent variables. This can be valuable when in the future the amount of information available about loan applicants is larger than there is time to address data issues like correlated variables. The research has also found added value of machine learning in terms of model performance. The Neural Networks and Random Forests produce more accurate results than (logistic) regression. The Support Vector Machines however are not suitable for retail credit risk predictions because the best predictions are made when models are trained with large amounts of data which proved to be problematic for the Support Vector Machines.
The results of this research depend on the Weight of Evidence transformation which is shown to be sub optimal for the Random Forests and possibly the other machine learning models. However while this transformation is suitable for Logistic Regression, the method is still outperformed by the Random Forests and Neural Networks.

Acknowledgements

Writing this thesis at the Financial Risk Management department of Deloitte has been a great experience during which I had the opportunity to learn a lot from the team members. They have shown great interest in my work and have helped me understand difficult topics, especially I would like to thank Florian Reuter and Martijn Westra who have monitored my weekly progress and contributed to my thesis by discussing their views on the subject with me. I would also like to thank my teachers Berend Roorda and Maurice van Keulen who guided me during the research and writing the thesis, their comments have helped me a lot and discussing my work at the university with them was always enjoyable and insightful.

During my time at the University of Twente I have had the tools to develop myself academically up to the point of obtaining a masters degree with this research, I would like to thank teachers as well as students with whom I have worked on projects for their involvement in this journey.

Furthermore my parents have always supported me during my time as a student, for which I am grateful. Also Rachelle van Liempt has been of great support in writing this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The first chapter introduces the research by describing the background and motivation for the research. The research questions are presented and the methods to find answers to these questions are introduced.

1.1 Research background

The article *Statistical Modeling: The Two Cultures* (Breiman, 2001b) describes two opposing views of predictive modeling. One assumes that data is generated by an existing stochastic process, the other is algorithmic modeling where the underlying mechanism generating data is assumed to be unknown/unknowable. The first view is an econometrics perspective and the second view is how data scientists approach predictive modeling. The author argues that the first culture has led to irrelevant theory and questionable scientific conclusions. The conclusions that are drawn from statistical modeling are about the model that is inferred from the data, and not on the real underlying process that generates the data. The primary concern of researchers and practitioners in the algorithmic modeling culture is with producing accurate results in terms of performance on data that was not used during the fitting of the models.
This research will compare common practice retail credit risk models, (logistic) regression, with three most dominant machine learning algorithms: Random Forests, Neural Networks and Support Vector Machines. Publicly available data from Lending Club will be used to create, validate and test the models. Lending Club is an American peer to peer lending platform that makes borrower information and historical loan performance publicly available. The company uses this information to assign risk grades to loans, and they encourage investors to slice and dice the data in order to find loan characteristics that they like and want to invest in. This encouragement has also attracted the attention of the scientific community. There are numerous articles published that study the Lending Club data set. The studies that incorporate machine learning, are concerned with minimizing defaults in order to avoid these during the selection of loans to invest in. In this setting false positives are much worse than false negatives. In other words a model recommending a loan that defaults is much worse than not recommending a loan that does not default. This results in models that are primarily good at finding the worst loans at the cost of misclassifying good loans. These models can be misleading in terms of the actual credit risk associated with an individual loan.
We will not take the side of the investor, but rather the side of a financial institution that needs to evaluate the credit risk of new loans. The fin-tech industry will keep challenging large financial institutions with new ideas and algorithms that enable them to provide better and cheaper financial services. From a risk management

perspective it is interesting to find out how good some of these algorithms are in constructing credit risk models on a large data set, however using these "black boxes" might be far from being accepted from a regulatory point of view.
Not only fin tech, but also big tech (Google Facebook) companies are potential entrants to the financial industry. These companies have access to personal information that could enable them to make highly accurate credit assessments. the financial institutions still have precious payment data of their clients, but to remain relevant in their industry they need to explore the methods used by data scientists to provide highly personalized and low cost financial services.

1.2 Research Questions

Machine learning and artificial intelligence have been changing our world over the past two decades, computers are trained to learn behavior of people in order to provide them with good product suggestions. Computers can even be trained to drive cars. As the learning algorithms become more sophisticated and advanced they are applied in a wider range of fields. The retail credit risk domain is a field where artificial intelligence has great potential, however the use of "black box" models is hard to explain in a regulatory context. Most credit risk models rely on a technique called logistic regression, one of the most basic models available in a data science toolbox. The question is, what do these newer and more advanced tools have to offer in the field of retail credit risk. The main question that we will answer in this research is formulated ad follows:

Where is the added value of machine learning in retail credit risk modeling?
To find added value, this research will focus on modeling approach and model performance of machine learning and (logistic) regression. The following subquestions have been formulated to be able to answer the main question in a structured way.

a) ***What are the current developments in retail credit risk?***
Answering this subquestion will help us understand where room for improvement currently is.

b) ***Which machine learning algorithms are suitable for credit risk modeling?***
Due to the huge amount of available models, it is necessary to select a few models to investigate in more detail.

c) ***How does traditional modeling differ from machine learning modeling?***
By creating the models, we can find possible added value in machine learning approach over logistic regression.

d) ***How do different algorithms perform on credit risk prediction?***
The last subquestion will be answered by evaluating model performance on data that was kept locked away during the creation of the models.

1.3 Methodology

The research methods will be structured as follows:

- *Literature research*
  To develop a theoretical framework that will answer subquestions a and b. The literature research will discus the following concepts.
  - Retail credit risk current developments
  - Machine Learning
    - Brief general machine learning introduction
    - Theory on the models that will be put in practice
  - Data transformation
  - Evaluation methods
- *Data*
  Publicly available credit data from LendingClub.com will be stored in a SQL database. To create a level playing field between models, a single data transformation is used to create a data set that all algorithms can work with.
- *Development of machine learning credit risk models*
  Different algorithms will be used, and for each algorithm different settings will be tested. the best performing model of every algorithm on every credit risk quantity is kept for evaluation.
- *Assessing model performance*
  Different performance metrics will be evaluated for the developed models. These performance metrics will be introduced in the next chapter.

1.4 Outline

In Chapter 2, the theoretical framework needed for the research, will described. In Chapter 3 the data set will be introduced to help the reader understand what data is available for our machine learning retail credit risk research and descriptive statistics will be provided. In Chapter 4 we apply algorithms discussed in Chapter 2 and develop machine learning credit scoring models. These models will be analyzed and compared in chapter 5. In the last chapter, Chapter 6, the conclusions drawn from the research will be presented and suggestions for further research will be discussed.

Chapter 2

Theoretical Framework

The theoretical framework will first discuss the current developments in retail credit risk and briefly explain more about retail credit risk. Then Machine learning and the specific algorithms that will be used in the research are explained in more detail. In the last part of the framework, data transformation and model evaluation will be addressed.

2.1 Current industry developments

Four aspects of developments in retail lending will be discussed, to provide insight in the relevant changes in the industry and for this research.

- *Peer to peer lending*

  The rise of Peer to Peer lending, where borrowers and investors are matched through an on-line platform, has been amplified by the financial crisis in 2008. People were excited to have access to credit lines without the need of traditional financial institutions that caused the crisis (Mateescu, 2015). What most people do not realize is that due to SEC regulations in the United States, the peer to peer loans have to be issued by a bank. The bank grants the loans to the borrowers and makes small securitized parts of the loans available to peer to peer investors. Peer to peer lending companies are only the market place where borrowers and investors can meet. These companies can charge a fee to investors because they assess creditworthiness of the potential borrowers and they make it easy for investors to find loans that match the investors' preferred characteristics.

- *Big data*

  An often heard buzzword is big data, but is it relevant for retail lending credit risk. The technological advances open doors to gather and store data that would have been impossible not too long ago. To assess creditworthiness of customers, models are created from historical data of consumer and loan characteristics, the model can then translate information of a loan applicant to a credit score. Having more and alternative data sources will give quantitative financial specialists more information about sources of risk in loan applications. However when dealing with truly big data traditional retail credit risk modeling might become problematic, if there are for instance more features than we have time to evaluate individually.

- *Consumer expectations*

  The consumer of today has high expectations, waiting for products or services is considered a thing of the past and everything should be accessible on-line at

the click of a button or through a convenient app on the smart-phone. When the consumer of today is not willing to wait for tangible products, it is not hard to imagine what is expected of intangible products like consumer credit. Another important factor is that consumers demand personalized products and services, they do not like having to accept a few standard options. These expectations and demands put pressure on the acceptance process of consumer credit and open the door for new technology in order to conform to the consumer expectations and demands.

- *Fintech*

  Financial technical startups pop up everywhere, which is problematic for the traditional financial institutions. The Fintech companies are able to start their business with state of the art technological equipment. They need less people to provide the same service more efficiently. These companies thrive on new technology and are passionate about using these technological developments to create better or cheaper financial products. An example of such a company is Advice Robo, they provide machine learning credit risk solutions to financial institutions.

2.2 Retail Credit Risk

In the lending business, credit risk is the main concern of the lender. It is defined as the risk of the situation where the borrowing counterparty fails to meet its financial obligation to the lender, resulting in full or partial loss of the funds invested by the lender. To compensate for these losses, lenders have to assess what loss is to be expected on a loan and charge that directly to the borrowers.

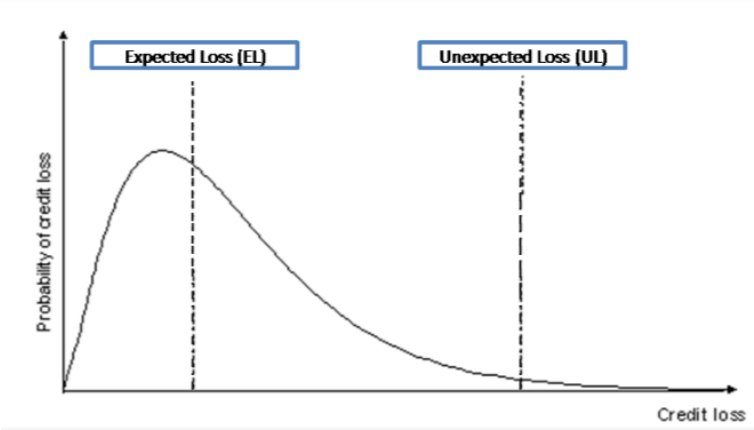


FIGURE 2.1: Loss Curve (riskarticles.com, 2017)

The unexpected loss is a measure of what the potential losses can be in a very adverse scenario, in this scenario much more borrowers fail to meet their obligations than they on average would. In regulatory context the unexpected loss is of interest in order to make sure a financial institution can survive a crisis.

2.2.1 Expected Loss

The expected loss (EL) is the product of the probability that a lender fails to meet its obligations, the degree to which the lender is exposed to the borrowers' failure and

the amount that can not be recovered from the borrower. These three quantities are called probability of default (PD), exposure at default (EAD) and loss given default (LGD).

$$EL = PD * EAD * LGD$$

Probability of Default

The foundation for the method to estimate this quantity has been laid more than half a century ago. It is called Logistic Regression and was first introduced in 1958 by David Cox. In the article he studies events with binary outcome dependent on multiple independent variables (Cox, 1958). The technique is nowadays still widely applied in the field of retail credit risk modeling. The method is about estimating the coefficients β of the following formula:

$$log\left(\frac{F(x)}{1-F(x)}\right) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

The left hand side is the called logit function, where $F(x)$ is the probability of default as a function of risk driving variables x. The right hand side of the equation is a linear combination of the different risk driving variables and their weights. In order for such a model to be qualified as a good model, a lot of model assumptions have to be satisfied.

Exposure At Default

For many products, the amount to which the bank or an investor is exposed in case of a default is known with certainty. However when dealing with unsecured amortizing loans (Lending Club), the exposure is not known in advance. When the exposure at time of default is uncertain, there is need for a model that estimates this quantity. Compared to PD and LGD the EAD is lagging behind in both industry practice and academic research (Qi, 2009). The most obvious model for amortizing loans would be a linear combination of loan amount and time until maturity. This method cannot be used in case the exposure at default must be estimated at loan application time. In that case a regression model is the standard approach.

Loss Given Default

The hardest credit risk quantity to model is the loss given default. The main reason for the difficulties in modeling loss given default is that default is not an absorbing state. A defaulted loan can 'cure', meaning that the borrower can repay the debt after a time of not being able to meet the obligations agreed upon when the loan was issued. When attempts to cure defaulted loans fail, lenders will try to recover as much as they can from the defaulted borrower. The percentage that can be recovered from the exposed amount is called the recovery rate (RR). It is common for recovery rate distributions to have high probability densities concentrated around lower and higher percentages, meaning that recovery attempts either have little effect or recover almost all of the exposed amount. The part of the exposed amount that can not be recovered is the loss given default, $LGD = 1 - RR$.

2.2.2 Charged Off loans

Because of the characteristics of the data set that we will introduce in Chapter 3, we will be using slightly different quantities to come up with the expected loss. These quantities are related to the event of a loan being charged off, this means that a third party is hired for the recovery process. The quantities of interest are called probability of charge off (PC), exposure at charge off (EAC) and loss given charge off (LGC).

$$EL = PC * EAC * LGC$$

2.3 Machine Learning

Machine learning algorithms, also referred to as statistical learning algorithms, performs tasks without being explicitly programmed. These algorithms can be separated in two forms, supervised and unsupervised machine learning. Supervised machine learning algorithms learn to predict an outcome, the response, based on the values of different features or variables. The data used to create a model with a supervised learning algorithm is historical and thus contains known response values. In contrast, unsupervised machine learning algorithms are models applied to data where there is no response value known. The performance of these models is hard to evaluate, because there are no observations to test predictions on. Unsupervised machine learning is used to learn relationships and find structure in data. Figure 2.2 gives an overview of some of the different algorithms that are used in machine learning context. The figure is not an exhaustive list of available techniques, but rather an example of how much is out there and that we need to narrow our research down to a few algorithms.

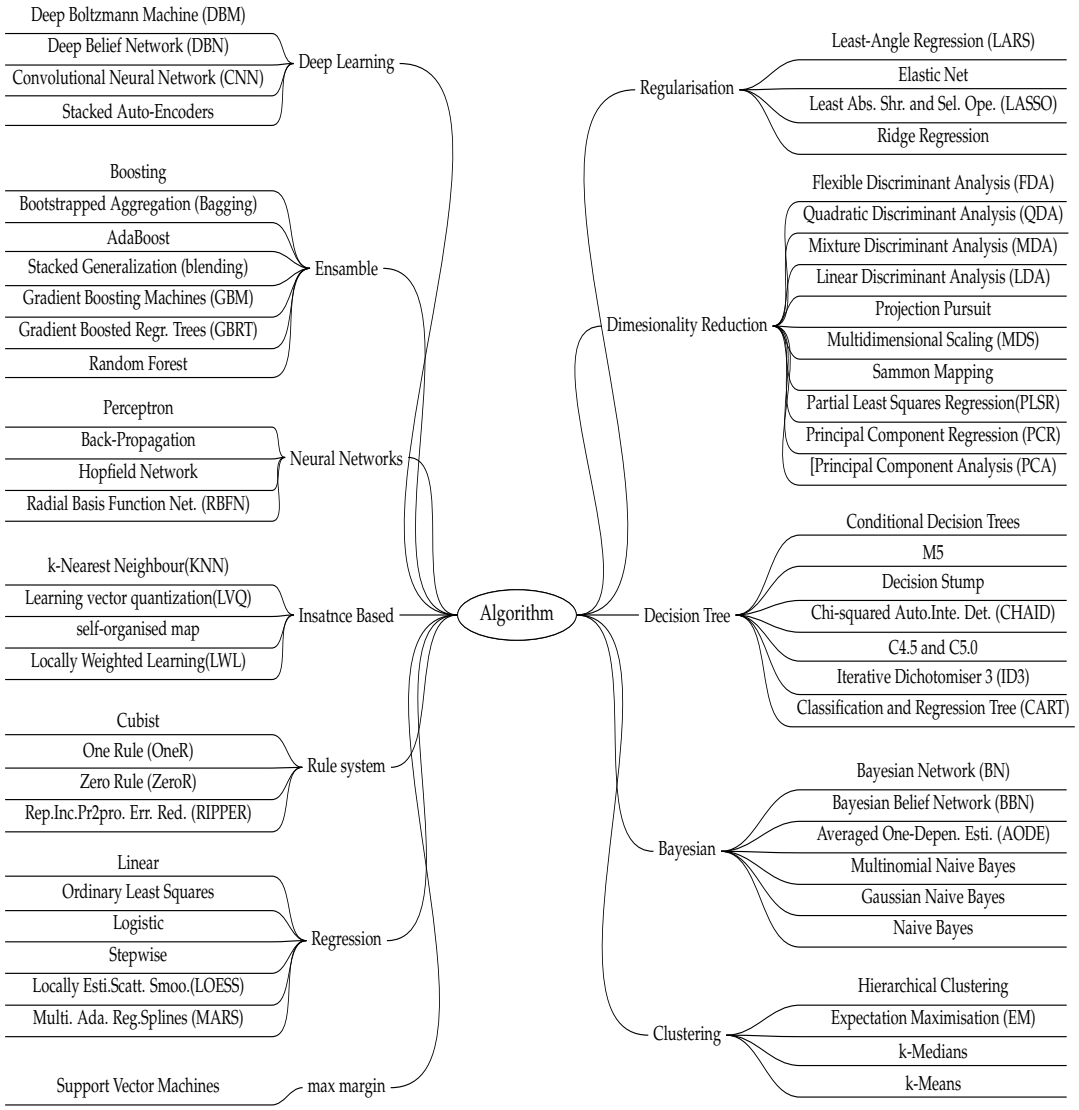


FIGURE 2.2: Machine learning algorithm mind map (Brownlee, 2013)

2.3.1 Algorithm selection

In our brief overview of machine learning theory, we focus on supervised machine learning algorithms. The data set, described in Chapter 3, that we use for machine learning credit risk modeling contains the responses for the quantities that we are interested in, therefore unsupervised learning would not be a sensible approach. Also because we want to predict a probability outcome and continuous outcomes we need to choose algorithms that are capable of handling classification as well as regression problems. Three supervised learning algorithms are selected that represent different machine learning streams.

- Random Forests
  An ensemble approach combining many weak models into a strong model.
- Neural Networks
  A structure resembling the brain, capable of learning complex relations in data.
- Support Vector Machines
  Mapping data into higher dimensional space to make it linearly separable.

These will be compared with the algorithm/technique that is currently most dominant in retail credit risk modeling.

- (Logistic) Regression
  Make predictions through a linear combination of input variables.

2.3.2 Random Forests

In this section, the Random Forest algorithm will be described. The building blocks of a Random Forest are individual decision trees, in particular CART which stands for *Classification And Regression Trees* (Breiman et al., 1984). These will be briefly explained, but we will not go into too much detail about the mechanics of individual decision trees.

The Random Forest algorithm (Breiman, 2001a) is an ensemble method where multiple predictors are combined into a single strong predictor. During the construction of a single tree in the forest, nodes are created to split observations, see figure 2.3 for example. At each node a preset number of candidate features are chosen from all features. The feature that can realize the best possible split is chosen from the subset of features, this process is repeated until the desired terminal node size is reached. Evaluating which feature realizes the best possible split can be performed with various methods, depending on the purpose of the tree (classification or regression), these will be discussed in the next subsections.

Randomness in the forest of decision trees is the result of two processes:

- Every tree is grown with a bootstrapped sample form the training data, a bootstrapped sample is a sample that is drawn with replacement from original data and contains the same number of observations as the original data set.
- Split decisions in the trees are chosen by evaluating all available features on their split performance. In terms of credit risk a good split would be separating all defaults from non-defaults. The random component is introduced by drawing random subsets of the available feature set.

When the number of features to chose from (mTry) is the same as the total number of features available and thus disregarding the second element of randomness, then we are essentially bagging. This technique significantly under performs Random Forests due to individual tree similarity. On the other hand, when there are 100 features and mTry is one, individual trees become very complex and out of sample model performance will be poor.

The terminal node size is another important parameter of a Random Forest, it is sometimes referred to as the depth parameter. A smaller terminal node size accounts for a more complex tree having more intermediate nodes, assuming that the number of observations used to train the forest is kept the same. The number specified for this parameter is the smallest node size that will be split by the algorithm.

It is important to realize that during the creation of a tree we start with all observations, and after a split we have two sets of observations. eventually we end up with n sets containing less than m observations, where m is the maximum size of the terminal nodes and n is the number of terminal nodes. When the tree is fully

grown, the response is determined in every terminal node which can be a value, a probability or a class. The prediction that a Random Forest makes is the aggregate of all tree predictions in the forest. All trees in a Random Forest have equal weight.

To summarize, a short overview of the Random Forest training process:

1. Split data into training and test sets.
2. Bootstrap a sample with the size of the training set, the observations are drawn with replacement.
3. Grow a tree for classification or regression using the bootstrapped sample.
4. Stop growing of the tree when minimal node size is reached .
5. Determine the response value in the terminal nodes.
6. Repeat steps 2 to 5 until the desired forest size is reached (number of trees in the forest).
7. For out of sample testing or predicting on new data, drop an observation down all trees in the forest, and aggregate the response values of the trees in the forest.

Classification

The Random Forest algorithm for classification, predicts a class that is best associated with an observation. An overview of a simple classification tree is given below. As described above such trees are combined into a forest and the predictions are aggregated. In terms of credit risk, a binary classification tree would use historical data to create trees that separate defaulted observations from non-defaulted observations.

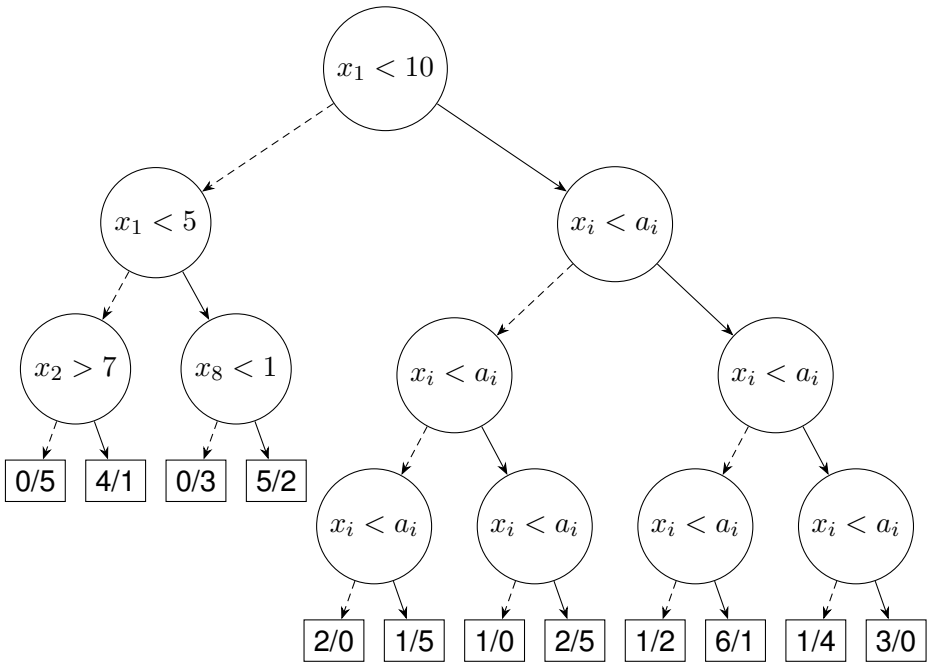


FIGURE 2.3: Example of a binary classification tree

The aggregation of individual tree predictions in a classification forest is done by selecting the majority vote. In binary classification a new observation is dropped down all trees, and either non-default or default is predicted depending on the majority vote. In Figure 2.3 the terminal nodes (rectangles) represent the distribution default/non-default. An observation with $x_1 = 2$ and $x_2 = 8$ will end up in the second terminal node from the left with default/non-default distribution $4/1$, this individual tree will predict the observation to default. The next step is the aggregation where the class that is predicted by most trees in the forest is the final forest prediction.

Probability Estimation

Probability estimation trees, so called PET's are almost identical to classification trees. The only difference is that the response value is a set of probabilities of an observation belonging to a class. Individual trees contain probabilities in their terminal nodes, if for example half of the observations in a terminal node belong to class 0 and the other half belongs to class 1, then the class probabilities in that terminal node are 50% and 50%. The class probabilities predicted by a probability estimation forest are the average class probabilities of all trees in the forest. According to (Malley et al., 2012) PET's are consistent probability estimators when their classification counterparts are consistent class predictors.

regression

Like a classification tree, a regression tree splits data into subsets until a certain depth of the tree is reached. In this case the terminal node value is the average of the response values in the terminal node. Figure 2.4 shows how this works. The figure represents a regression problem with two independent variables and one response variable. Every split is represented by a line in the scatter-plot, the rectangles are the terminal nodes of the regression tree. Splits are chosen in such a way that the variance inside the child nodes is smallest. The values at the end of the tree correspond with the values inside the rectangles, they are the average of the response variables.

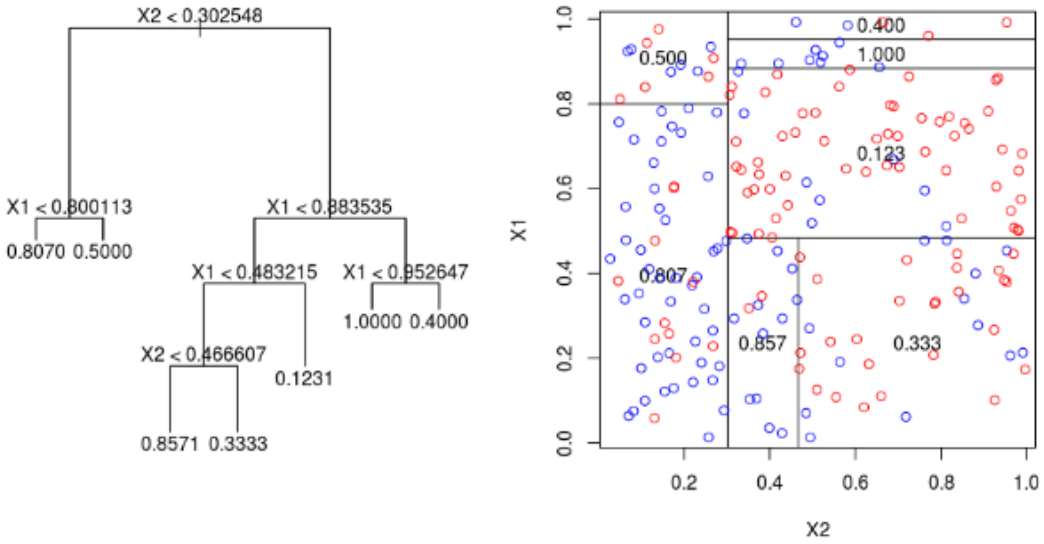


FIGURE 2.4: Regression Tree visualization (Charpentier, 2013)

The prediction of a regression forest is like a probability estimation forest, the average of all tree predictions.

Variable importance classification/probability estimation forests

After a Random Forest is trained, it is possible to plot the importance of the variables used in the model. The method to evaluate the importance of a single variable is to calculate the *mean decrease Gini*. The original paper about the Gini coefficient, Variabilità e mutabilità (Gini, 1912), was intended to measure income equality of a country. Today it is still used to measure inequality. A low Gini coefficient represents more equality in a data set. Now if we think about the trees in a Random Forest, every node contains a certain distribution of the response variable. After a split, the child nodes should have a lower Gini coefficient, because the goal of the splits is to make the class distributions in the child nodes as pure as possible, all observations should have the same class. So the variable that was used to make the split has decreased the Gini. If we calculate the decrease in Gini for all variables and all trees in the forest, we can find the mean decrease Gini for every variable. The most important variables for a forest are those that have the highest mean decrease Gini.

Variable importance regression forests

For regression forests it is also possible to assess the importance of the model variables, the measure used is *mean decrease accuracy*. The out of bag (OOB) observations, these are the observations that are left out in the bootstrapping process, are used to calculate this variable importance measure. After the creation of a tree, OOB samples are dropped down the tree, and the prediction accuracy is recorded (Friedman, Hastie, and Tibshirani, 2001). Then the decrease in accuracy can be calculated for a single variable by replacing the values of said variable with a random permutation in the OOB data, resulting in a decrease in accuracy that is averaged over all trees.

2.3.3 Neural Networks

Neural Networks consist of nodes and edges, they are inspired by neurons and synapses of a brain. The synapses or edges transport information to the neurons or nodes. The nodes in a Neural Network collect information from the synapses and transform the information with an activation function. Eventually the information will arrive at an output node, or multiple output nodes. These output nodes contain the response that is predicted based on the input values.

For a simple Neural Network without hidden layers and a logistic activation function, we can calculate the output with the following formula:

$$y = f\left(w_0 + \sum_{i=1}^{n} w_i * X_i\right)$$

In the formula, f represents the logistic transformation. As we can see this is the same as a Logistic Regression, however Neural Networks become more interesting when hidden layers are introduced. These layers allow for nonlinear combinations and more complex relations in the data to be captured by the Neural Network. In this research we will use "vanilla" Neural Networks (Friedman, Hastie, and Tibshirani, 2001), which are networks with one hidden layer and a logistic activation function.
The 'extra' nodes at the top of the network, which can be seen in Figure 2.5, contain a constant input of one and are called bias nodes. These nodes can be seen as the intercept term in Generalized Linear Models (GLM's), the family where Logistic Regression belongs to.

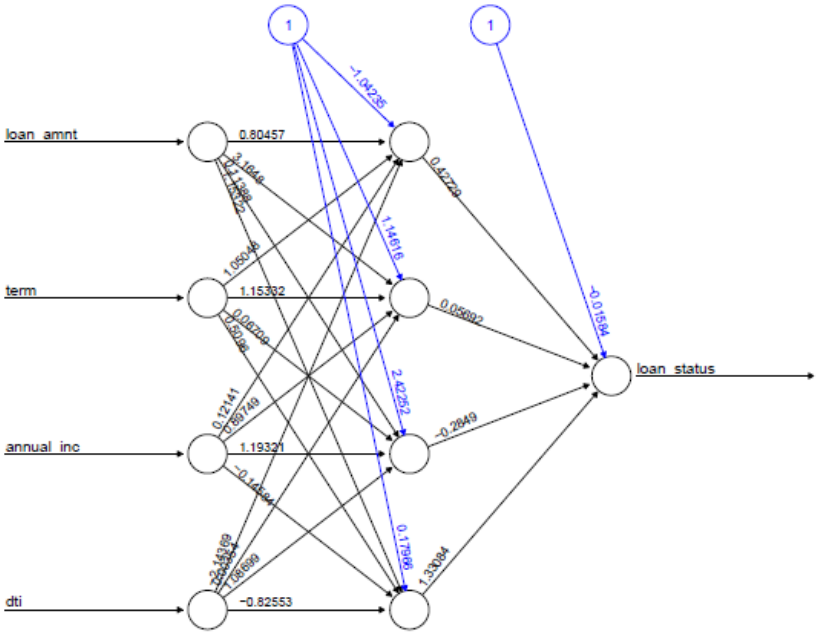


FIGURE 2.5: Example of a small Neural Network

This simple network has four input neurons, one hidden layer with four neurons and one output neuron. The values on the edges in the network, are the weights that the algorithm learns during the training of the network. Unlike in GLM's it is

not easy to observe how a change in one input variable is related to different output, because the change in output can also be related to interaction terms in the network.

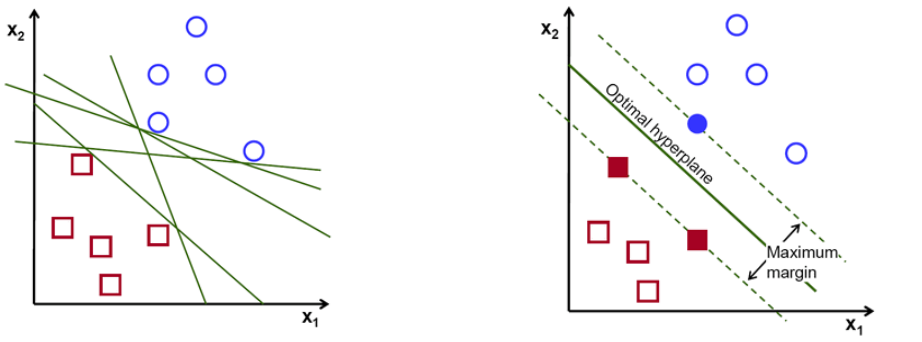
Backpropagation

At the beginning of the training of a Neural Network, the weights are chosen at random. Then using the training data a forward pass through the network is executed and the prediction error is calculated. Obviously this error should be minimized, this is done by a number of iterations of forward and backward passes through the network. A backward pass is calculating the derivative with respect to the error in every point/weight of the network. Before a new iteration, all weights are updated according to a beforehand specified learning rate and the derivatives. The training of the network is finished when the weights have converged and the error does not decrease anymore. A mathematical explanation of the backpropagation algorithm can be found in the book; "Elements of statistical learning" (Friedman, Hastie, and Tibshirani, 2001).

2.3.4 Support Vector Machines

Support Vector Machines, as introduced by Vladimir Vapnik in "The Nature of Statistical Learning Theory" 1995, also called large margin classifiers, try to find an optimal separating hyperplane to determine to which class an observation belongs.

classification



(A) Separating hyperplanes

(B) Optimal separating hyperplane

FIGURE 2.6: separating hyperplanes (OpenCV, 2017)

The data in Figure 2.6 (A) can be separated in different ways, all of the green lines separate the data perfectly, intuitively none of these hyperplanes feels like the best classifier. In Figure 2.6 (B) we see the optimal classifier, in this case the margins between observations and the hyperplane are largest, the shortest line from the observations on the margins to the hyperplane are called the support vectors, hence the name Support Vector Machines. In this simple example the data can be separated, in larger and more complex data sets this will be rare. To solve this problem, two steps can be taken individually or both at the same time. A transformation can be applied to attempt separating the data in a non linear feature space, and soft margins can be applied. The non linear approach is called the kernel trick, different kernels are available to find a feature space where data can be separated. The second option, a soft margin, allows the model to leave observations in the margin or even at the

wrong side of the hyperplane. Through introducing slack variables, a Support Vector Machine can have a soft margin.

The mathematical formulation of finding the maximum margin classifier is as follows:

$$
\begin{aligned}
&\max_{\beta_j, \epsilon_i} M \\
&subject\ to: \\
&\sum_{j=1}^{p} \beta_j^2 = 1, \\
&y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \\
&\epsilon_i \geq 0, \\
&\sum_{i=1}^{n} \epsilon_i \leq C
\end{aligned}
$$

The slack variables ϵ indicate if an observation was on the good side of the hyperplane, $\epsilon = 0$ or on the wrong side of the hyperplane, $\epsilon > 1$. Parameter C is a constraint on how big the slack variables can become. The problem formulation also includes multiplying M with $(1 - \epsilon_i)$ which lowers the value of M when observations are on the wrong side. The y_i represents the class label which can take a value of -1 or 1. A more detailed explanation Support Vector Machine mathematics can be found in the book; *An introduction to statistical learning* (James et al., 2013).

kernels

The examples in figures 2.6 and 2.8, are cases of Support Vector Machines with a linear kernel. Other kernels can be used used to map the input to a higher dimensional space in which a linear solution can be found. Kernels have to satisfy Mercers theorem (Korotkov, 2011), we will not go into this theorem because it goes to deep for the high level analysis of machine learning in this research. Figure 2.7 shows what the goal of a kernel is.

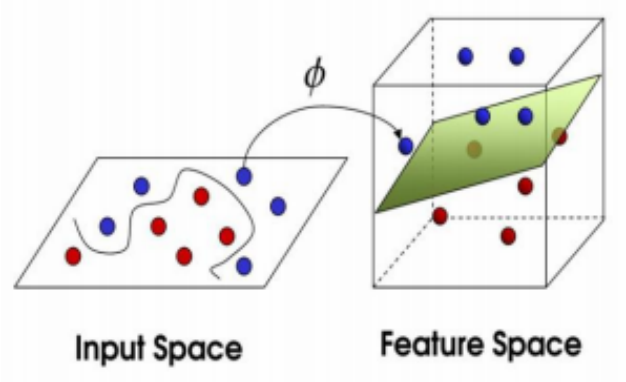


FIGURE 2.7: Mapping input space to feature space (Raghava, 2006)

The SVM optimization problem can be solved by using only the inner product of all training observations. The inner product of two vectors a and b, denoted by $\langle a, b\rangle$, in an n dimensional space is obtained by $\sum_{i=1}^{n} a_i b_i$. Kernels are functions that transform these inner products.

The available kernels are:

- Linear
  $K(a, b) = \langle a, b\rangle$
- Polynomial
  $K(a, b) = (1 + \langle a, b\rangle)^d$
- Sigmoid
  $K(a, b) = tanh(\gamma\langle a, b\rangle + 1)$
- Radial
  $K(a, b) = e^{-\gamma\langle a,b\rangle^2}$

The influence regions of the sigmoid and radial kernels are controlled by parameter γ, the polynomial kernel can be of different degrees d and the linear kernel is computed with only the inner product of the vectors belonging to training observations.

Probability estimation

To obtain a probability estimate from the binary classification Support Vector Machine, a logistic model is fitted to the decision values (Meyer et al., 2017). With the logistic model, the decision values can be transformed to probabilities between zero and one.

Regression

Similar to classification the Support Vector Regression algorithm finds margins. The big difference is that the observations should be inside the margins of the support vectors. Figure 2.8 is an example of Support Vector Regression. The non support vectors do not contribute anything to the model, the regression line is the result from the choice of margin and the support vectors.

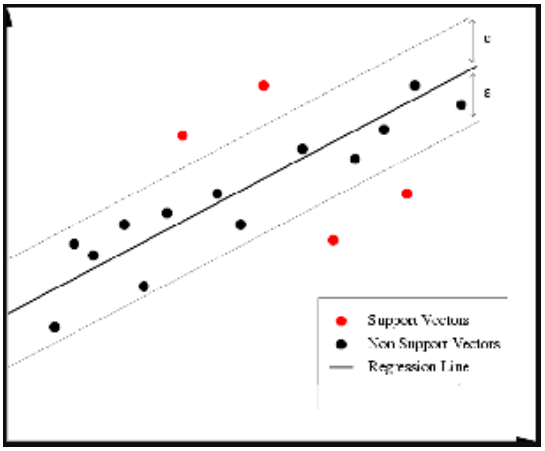


FIGURE 2.8: Support Vector Regression (Gilardi and Bengio, 2000)

A big difference with ordinary least squares regression is that the support vector method uses the shortest distance to the regression line/margins, where ordinary least squares regression uses the vertical distance to find the line with the lowest value of the sum of squared errors.

2.4 Data transformation

In most cases, raw data is unsuitable for modeling. The major issues that prevent raw data to be used as input for models are:

- Missing values
- Non numeric variables
- Outliers

To overcome these problems, we have a few standard options. For instance replacing missing values with the average of the non missing values, creating binary dummy variables for a categorical field and disregarding observations that are outside a number of standard deviations of the mean to make sure there are no unusually low or high values that have high impact on the outcome of a model.

Another reason for a data transformation is that machine learning algorithms in most cases benefit from having input that is on the same scale. With Logistic Regression for example scaled data makes the coefficients directly interpretable, a higher coefficient translates to a higher importance of that variable in the model.

Generalized linear models (GLM) are linear regressions where a link function is used to transform output to make sure it has the right characteristics (Nelder and Wedderburn, 1972). When probability is estimated, the output must be in the interval $[0, 1]$ to achieve this a logistic transformation is often used. Another option, in this case, would be the probit transformation using the cumulative density function of a standard normal distribution having some advantages when a normal prior distribution is assumed.
When the logistic transformation is chosen, data can beforehand be transformed, by means of Weight of Evidence (WOE). This method uses the following formula and binned data to assign weights B_i in terms of the log odds ratio of the binary response variable.

$$WOE_i = log\left(\frac{P(B_i|Y=1)}{P(B_i|Y=0)}\right)$$

After this transformation, all variables have the property that a higher WOE bin value corresponds to a higher probability of $Y = 1$

The WOE method takes categorical data fields as a bin per level of a category and missing values are handled as a separate level. The latter has the advantage that a field, left blank, by a consumer can possibly hold more information than replacing the blanks with the average value for that field.

Simple WOE example with one categorical variable:

Home	Status
OWN	D
RENT	D
OWN	ND
RENT	D
NA	ND
NA	D
OWN	ND
RENT	ND
OWN	D
RENT	ND
OWN	ND
OWN	ND
RENT	D

TABLE 2.1: Observed loan status of borrowers with home information

	D	ND	WOE
OWN	2	4	-0.5390
RENT	3	2	0.5596
NA	1	1	0.1542

TABLE 2.2: Default frequency per level and corresponding WOE

In Table 2.1 we see borrowers that either own or rent a home and two observations have missing values indicated with NA. The borrowers can be in default (D) or not (ND). Table 2.2 shows the frequencies of the three levels of the categorical variable and their corresponding WOE values calculated with the WOE formula.

In Logistic Regression, we try to find a linear combination of the variables on which we use a logistic transformation to find a probability of belonging to class $Y = 1$. Logistic Regression and Weight of Evidence are a strong combination because the Weight of Evidence transforms variables to have a linear relationship with the logistic function used in Logistic Regression. The Neural Networks that we use have logistic activation functions, therefore Weight of Evidence can also be beneficial. With Support Vector Machines and Random Forests, we have less reason to believe that Weight of Evidence is a good transformation. However for this research the creation of a level playing field by choosing one data transformation for all models justifies choosing the Weight of Evidence approach.

2.5 Evaluation Methods

Algorithms described in this chapter will be evaluated with the following methods:

- Probabilities
  Binary classification probability estimates will be evaluated with the receiver operating characteristic.
- Continuous quantities
  These will be evaluated with Mean Squared Error and R squared. Expected Loss predictions, which are continuous, are also evaluated in terms of ability to capture observed losses.

2.5.1 Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC) is a tool to measure the discriminatory power of a binary classification model. It is a plot of the true positive rate versus the false positive rate of model predictions at different cut off values. The area under the curve (AUC) of the ROC is often used to quantify discriminatory power of a model in one number.

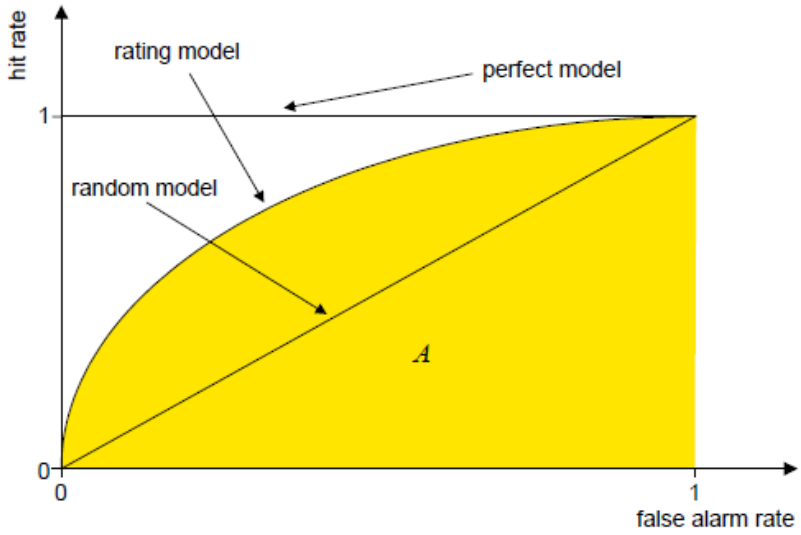


FIGURE 2.9: Receiver Operating Characteristic curves (BCBS, 2005)

Figure 2.9 shows three ROC 'curves'. The straight line starting from the origin is the performance of a random model, the performance of a rating model and the performance of a perfect model. Any rating model should operate between the performance of a random model where $AUC = 0.5$ and a perfect model where $AUC = 1$

If model PD of an observation belonging to the defaulted population is denoted PD_D, and the PD of an observation belonging to the non-defaulted population, PD_{ND}. Then the AUC can be interpreted as $P(PD_D > PD_{ND})$ when from both populations one observation is randomly drawn (BCBS, 2005).

2.5.2 Mean Squared Error

The Mean Squared Error (MSE) of predictions compared to actual outcomes in the data set, calculated with the following formula:

$$MSE = \frac{1}{n} \sum (\hat{Y}_i - Y_i)^2$$

The errors $(\hat{Y}_i - Y_i)$ in the formula are the predicted minus the observed values. When these are squared and divided by the total number of observations, we obtain the MSE. A model with an MSE of zero is a perfect model because it always predicts the actually observed values.

2.5.3 R squared

The R^2 is the amount of variance in a response variable that can be explained by the model, calculated with the following formula:

$$R^2 = 1 - \frac{\sum (\hat{y}_i - y_i)^2}{\sum (\bar{y} - y_i)^2}$$

The top part in the fraction is the summed squared residual of predicted minus observed values. The lower part of the fraction is the summed squared residual of average observed values minus observed value. A perfect model can account for all variance in the response variable and achieves an R^2 of 1. A model which can account for none of the variance in data achieves an R^2 of 0.

2.5.4 Loss capture ratio

Here we rank the observations on risk prediction in descending order and plot the corresponding observed losses. This method is visually similar to the ROC, the curves are all starting in the origin and moving to (1,1). In this case we find the number of observations as a percentage on the horizontal axis and on the vertical axis we find the cumulative observed loss. An optimal model will rank all losses in descending order of their severity, this will be a steep line rising to 100% of the losses and from that point the optimal model will move to (1,1) parallel to the x axis. A model that comes close to the described optimal line is desired. When a model comes closer to the line from the origin to (1,1) it means that the model is not much better than a random ranking of the observations.

Chapter 3

Lending Club data set

In Chapter 3, the data set that is used in the research will be described. First we will introduce Lending Club, the company that has made the data available, and then we will discuss what data is available and how we will prepare and create different data sets that will be used to create and analyze different credit risk models.

3.1 Company Profile

Lending club is an award winning peer to peer lending platform. One of its goals is to change the lending industry by providing better rates than traditional financial institutions and by operating at much lower cost. Lending club is serving as a platform where borrowers and investors find each other. Most of the borrowers are individuals who want to finance other loans that have adverse rates, for a more favorable rate through lending club. Figure 3.1 from the Lending Club website gives a percentage overview of the most common loan purposes.

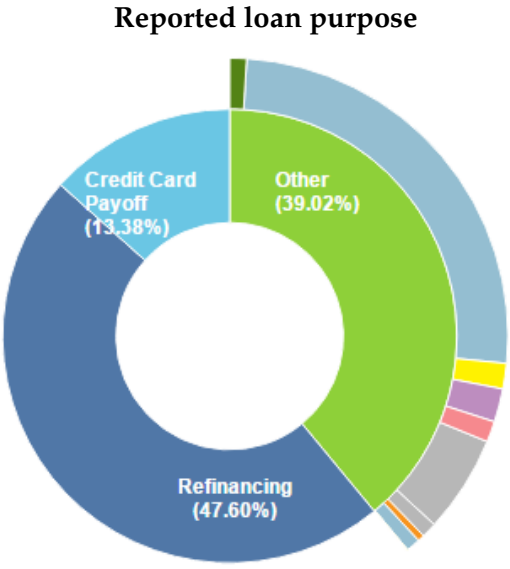


FIGURE 3.1: Most common loan purposes as percentage of all loans (LendingClub, 2016)

The figure shows that more than half of the loans issued by Lending Club are related to other debt that the customers already have. To give an overview of the purposes that borrowers can choose from, the purposes are listed on the next page.

Credit card	House	Debt consolidation
Small business	Educational	Medical
Other	Vacation	Moving
Wedding	Home improvement	Major purchase
Car	Renewable energy	

The investors on the other side of the platform are people that want to have better returns than they can find in more traditional investments like the stock market or on deposit accounts.

"Don't take our word for it. See for yourself. Our entire loan database is available to download. Help yourself to our data, and slice and dice it anyway you want. Try that at your favorite banking institution!"

-Lendingclub.com

They are encouraged, by Lending Club, to explore the data in order to find characteristics of borrowers that suit their investment strategy. To help investors that do not want to go through the extensive amount of data available, Lending Club provides risk grades per loan, which are also translated into interest rates on loans.

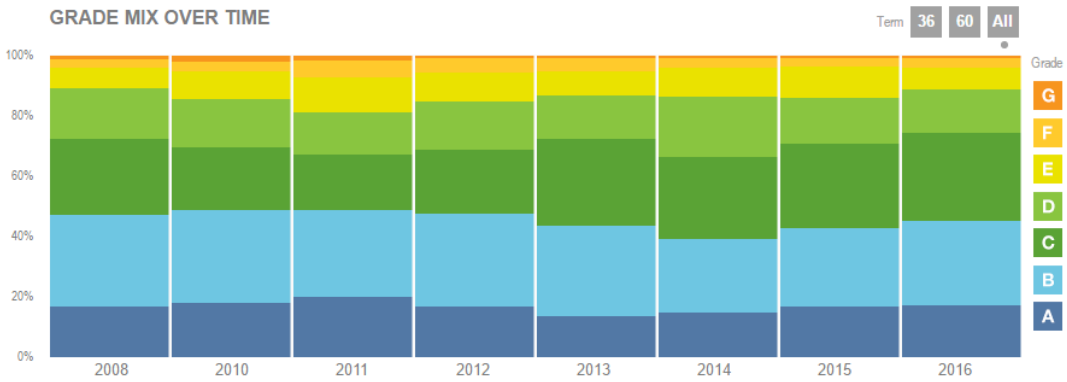


FIGURE 3.2: Lending Club grade mix over time (LendingClub, 2016)

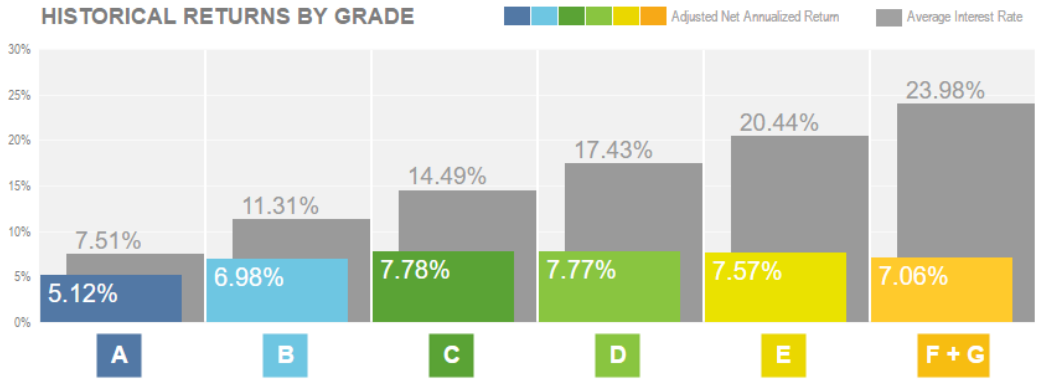


FIGURE 3.3: Lending Club annualized net returns per risk grade (LendingClub, 2016)

There are seven risk grades and each grade has 5 sub grades resulting in 35 distinct risk levels that can be assigned to a loan. In Figure 3.2 the distribution of grades assigned to loans over the past years is provided, notice that around 70% of all loans are in the top three risk grades, indicating that Lending Club accepts more relatively safe borrowers than the riskier borrowers. Figure 3.3 gives an overview of the adjusted net annualized returns per sub grade assigned by Lending Club. The adjustment in this calculation is for expected future losses.

3.2 Data Set

The data used for this research was downloaded in September 2016. Lending Club has made the data available in six comma separated value (.csv) files. Also a file containing the complete payment history of all loans is available. Table 3.1 presents the available loan data files:

File	Loans
2007-2011_LoanStats3a	39,786
2012-2013_LoanStats3b	188,181
2014_LoanStats3c	235,629
2015_LoanStats3d	421,095
2016Q1_LoanStats_2016Q1	133,887
2016Q2_LoanStats_2016Q2	97,854
total	1,116,432

TABLE 3.1: Lending Club data files

The size of the different lending Club files shows the growth that the platform has gone through since its origination. Currently more than 400 thousand loans are funded every year making increasingly more data available for credit risk research.

Combining these files results in a database of 1,116,432 loans with 111 columns that contain information about the loans. In this data set we observe loans with different statuses, shown in Table 3.2:

loan_status	observations	percentage
Fully Paid	332,844	29.813%
Default	85	0.008%
Charged Off	78,627	7.043%
Current	673,327	60.311%
In Grace Period	9,792	0.877%
Late (16-30 days)	4,574	0.410%
Late (31-120 days)	17,183	1.539%
Total	1,116,432	100%

TABLE 3.2: Loan statuses in the Lending Club data

There are almost no loans labeled 'Default' in the data, and more than half of the loans have status 'Current'. We will proceed with loans that are either fully paid or charged off. We are interested in supervised learning, therefore we only keep loans

that have matured. Loans that are in grace period, late or in default reside in non absorbing states, and are not taken into account. Would there have been a more significant amount of defaults, then we could take them into account in predictive modeling. This decision is a result of Lending Club's policy to charge off loans very fast in stead of managing a portfolio of defaulted loans for their investors.

The data set after removal of loans not in absorbing states is shown in Table 3.3:

loan_status	observations	percentage
Fully Paid	332,844	80.981%
Charged Off	78,627	19.109%
Total	411,471	100%

TABLE 3.3: Loan statuses of matured loans

The payment history data set contains valuable American credit information called the FICO score, this score is a credit rating for consumers. Lending Club is currently not providing the FICO score directly with the other loan information, they have changed their open and transparent strategy by making it harder for investors and researchers to obtain all data. On the contrary, Lending Club has also added data fields in the past years, some at the request of the community. Coming back to the FICO score, this data is added to the matured loan information data by joining on loan id using an SQL database.

implications of disregarding non matured loans
By having two possible states of a loan, Fully Paid and Charged Off, we are dealing in a different than usual manner with expected loss. We will model the Probability of Charge Off (PC), Exposure at Charge Off (EAC) and Loss Given Charge Off (LGC). The advantage that we get from this is that we do not need to deal with the possibility that a default cures, as described in Section 2.2.1. The quantities EAC and LGC will be modeled using only the charged off observations.

3.2.1 Descriptive statistics

Table 3.4 provides descriptive statistics of a small set of the numeric features of matured Lending Club loans that will be used in the models. A full list of descriptive statistics for the numerical features can be found in Appendix C. The second column of the table contains the number of observations on which the descriptive statistics were calculated. For a lot of variables, this number is lower than the number of observations in our data set. This is mainly the result of Lending Club adding features over time. The oldest data set available contains the variables that have 411471 observations present in Table 3.4. The models we use in Chapter 4 do not need to be prepared to handle missing values, because we perform a weight of evidence transformation on our data that will handle the missing values. A complete description to understand the names of the more ambiguous variables is provided in Appendix A.

Variable name	n	mean	sd	min	max	range
loan_amnt	411,471	13,955.73	8,275.54	500.00	40,000.00	39,500.00
funded_amnt	411,471	13,929.23	8,261.56	500.00	40,000.00	39,500.00
funded_amnt_inv	411,471	13,868.81	8,273.50	0.00	40,000.00	40,000.00
term	411,471	41.60	10.15	36.00	60.00	24.00
annual_inc	411471	73623.27	62539.79	0.00	8,900,060.00	8,900,060.00
dti	411,471	17.19	17.70	0.00	9,999.00	9,999.00

TABLE 3.4: Descriptive statistics

The table shows a few interesting things. The first being that the amount funded is not alway the same as the amount funded by investors, there are even loans that have an amount invested by investors of zero. Lending Club sometimes also invests in loans to get them funded resulting in this difference. Another curious thing is that the data set includes observations of loans that are provided to individuals that have no income and an individual with an annual income of 8.9 million. With a maximum loan amount of 40 thousand it seems highly unlikely that someone earning multiple millions per year takes out a loan for a few thousand dollars. These peculiarities are smoothed away by the WOE transformation described in Section 2.4.

3.3 Credit Risk Evaluation

In the empirical analysis performed by (Emekter et al., 2015) the available data until 2011 is analyzed. On matured loans, a logistic regression is performed, to predict defaults. They found that the Lending Club credit grades (as dummy variables) were the only significant variables. Thereby verifying that the lending Club sub grades adequately predict credit risk, with the exception that the F grade was riskier than the lower G grade. Their research reports 18.6% defaults in the set of matured loans, this is close to the 19.1% from Table 3.3 that we observe in our more recent data set.

To provide a overview of the credit risk associated with investing in the different Lending Club sub grades, we have calculated the historical averages of received interest, Charge off rate, Exposure at charge off, loss given charge off and percentage of principal lost. These can be found in Table 3.5.

sub grade	interest	Charge off rate	EAC	LGC	% of principal lost
A1	6.19%	2.89%	84.97%	91.90%	2.25%
A2	6.98%	4.52%	75.73%	91.96%	3.15%
A3	8.08%	5.54%	68.81%	92.45%	3.52%
A4	8.58%	7.01%	73.80%	92.22%	4.77%
A5	9.32%	8.54%	74.86%	92.64%	5.92%
B1	10.55%	9.56%	81.57%	92.41%	7.20%
B2	11.85%	10.50%	83.54%	92.19%	8.09%
B3	13.21%	12.21%	66.70%	92.59%	7.54%
B4	13.73%	13.52%	76.99%	92.18%	9.59%
B5	13.59%	15.52%	76.09%	92.26%	10.90%
C1	14.22%	17.35%	79.30%	92.48%	12.72%
C2	14.77%	18.95%	81.47%	92.57%	14.29%
C3	15.02%	21.13%	72.01%	92.33%	14.05%
C4	15.41%	22.96%	70.32%	92.15%	14.88%
C5	16.22%	24.13%	78.15%	92.29%	17.40%
D1	16.66%	25.72%	62.04%	92.81%	14.81%
D2	17.84%	26.95%	75.39%	92.17%	18.72%
D3	18.11%	27.31%	64.57%	92.31%	16.28%
D4	18.51%	30.07%	84.22%	92.09%	23.32%
D5	19.10%	30.95%	85.24%	91.80%	24.21%
E1	18.57%	33.40%	84.10%	91.76%	25.78%
E2	19.66%	35.26%	65.36%	92.13%	21.23%
E3	19.75%	36.52%	87.13%	92.63%	29.48%
E4	20.91%	37.98%	64.14%	92.24%	22.47%
E5	21.09%	39.15%	86.63%	91.44%	31.01%
F1	22.45%	38.42%	63.30%	94.19%	22.91%
F2	22.65%	41.53%	63.23%	92.40%	24.27%
F3	23.56%	42.97%	64.60%	92.80%	25.76%
F4	23.11%	45.33%	83.71%	92.43%	35.08%
F5	23.08%	46.98%	86.53%	92.10%	37.43%
G1	23.71%	46.02%	62.27%	92.88%	26.62%
G3	23.73%	48.75%	90.81%	91.95%	40.71%
G4	24.45%	40.37%	89.06%	92.53%	33.27%
G5	21.12%	49.69%	88.12%	92.07%	40.31%

TABLE 3.5: Overview of credit risk per Lending Club sub grade

The grades below C have a high average percentage of principal lost, However in Figure 3.3 Lending club reports a return on investment which is about 7% for all grades, except for the highest grade where the return on investment is 5.12% on average. These high losses can be covered by the interest rates because the interest rates in the table are annualized rates and the total average interest earned is therefore higher than the average percentage of principal lost. Furthermore we see in Table 3.5 that the charge off rates increase with the sub grades, the average exposure at charge off fluctuates between 60% and 90% and that the loss given charge off is quite stable across the different sub grades with an average of 92.3%. We can already see that the impact of a Loss given charge off model over just taking the average will be limited because most charged off loans will have a loss given charge off close to 92.3%.

3.3.1 Predictive power individual features

By looking at the density plots of charged off loans versus fully paid loans on every variable, we can conclude that almost all features contain very little information about the status of a loan. These plots are added in Appendix B. When the predictive power of individual features is low and there are a lot of features, machine learning is a promising approach. The algorithms are suitable for handling a large amount of features and finding patterns that linear models cannot discover.

3.4 Data preparation

Not all variables are eligible for the kind of predictive modeling that we aim to do, as (Tsai, Ramiah, and Singh, 2014) describe in their lessons learned section. Some of the variables in the data set are not known at application time, including these in the models will allow the models to cheat by looking into the future. Therefore, the following features are excluded:

loan status	The response variable
out prncp	The exposed amount at time of maturity, larger than zero means charged off
out prncp inv	Same as out prncp, however can be smaller when lending club and other investors co financed the loan
total pymnt	Payments received on the loan
total pymnt inv	Proportion of payments received on the loan for investors
total rec int	Received interest, not available at time of loan application
last pymnt d	Payment data, not available at loan application
last pymnt amnt	Payment data, not available at loan application
last credit pull d	Last date that credit information was requested on the borrower, not available at loan application

Features excluded for different reasons are:

id	Identification number of loan, not useful for prediction
member id	Identification number of member, not useful for prediction
int rate	Directly linked to Lending Club risk prediction
installment	Monthly amount to be paid, also linked to LC risk prediction
initial list status	Was the loan intended for wholesale or fractional sale
grade	Lending Club risk grade
sub grade	Lending Club risk sub grade
emp title	Name of the employer of a borrower, text field
issue d	Date variable
pymnt plan	Logical indicating if payment plan is arranged
policy code	Was the loan publicly available, removed by lending club
url	Link to loan listing on LC website, text field
desc	Additional loan description, text field
title	Name of the loan, text field
earliest cr line	Date variable

To handle categorical data we can make dummy variables, unfortunately there are a lot of categories and transforming all categorical variables to dummy variables blows up the amount of variables in the data set. Alternatively the Weight of Evidence method described in Chapter 2 is used. This method is also applied to numerical data by binning the these variables. The R "information" package (Kim, 2016) calculates the WOE scores. The transformation is applied after the data is split in different sets for training a model, validating the model and later testing the model, this will be motivated in Section 3.6. When the data is eventually transformed into WOE scores we have to be careful with direct interpretation of the scores, because a high score on a variable can be the result of correlation with another variable. In short this means that conditional independence should be satisfied to draw conclusions about the WOE score of a bin or category.

3.4.1 Data Skewness

In classification problems its is generally accepted that balancing a data set is good practice and will lead to better results. From Section 3.2 we know that the data is 80/20 percent skewed with respect to the fully paid and charged off loans. Keeping the original distribution in the data will result in probability of charge off estimates in line with historical data. Using a balanced set would imply that the historical distribution of fully paid and charged off loans is not representative of the new data, and that charged off and fully repaid loans are equally likely. Secondly the imbalance of 4:1 would only be problematic if there is little data available. Because the distribution in the train and test data will be the same and there is enough data, the data set is not balanced.

3.5 Adding features

At loan application, borrowers have the option to provide more details about their loan request. They can provide the name of their employer and add additional description of loan purpose. An interesting feature to add might be a logical variable indicating whether the employer name or additional description was provided.

provide emp name	binary variable indicating whether the employer name was provided
provide description	binary variable indicating whether the loan description was provided
diff fundinvfund	when Lending Club invests in a loan to make sure it gets funded, there is a difference in the amount funded and the amount funded by investors, see Section 3.2.1
issue month	extracted from the issue date variable, to be able to capture seasonality influences in our models

To check if these features add predictive power, their distributions over the two loan statuses are plotted in Figure 3.4. In the plots, status 0 corresponds to fully repaid and status 1 to charged off.

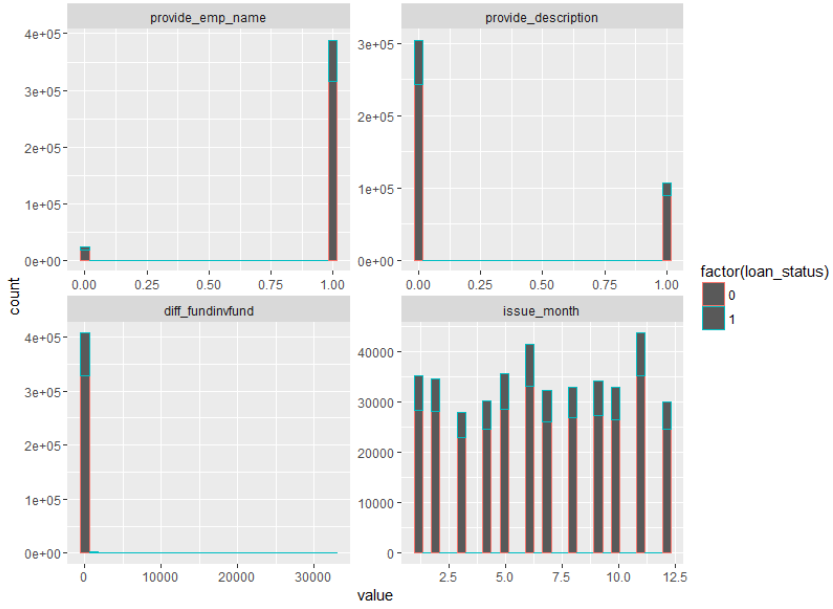


FIGURE 3.4: Distribution of new features over different loan statuses

Like the original features in the data set, they seem to have little predictive power on their own. However the added features could prove to be valuable in nonlinear models.

3.6 Creating data sets

For the development of all models we split the number of observations that we will use into a train and validation set of 70% and 30%. These two sets will be created from 400 thousand loans or a smaller subset of those 400 thousand loans. The 11471 loans that we have left are saved for testing the models in Chapter 5. The distribution of data over all sets is then 68% in the train set, 29% in the validation set and 3% in the test set. It might be expected that the validation and test data sets would have the same amount of observations, in this research we will focus on training and validating the models with large amounts of data and compare the models with the performance on a smaller set. The size of the relatively small test set is however substantial enough to be used for comparing the different models. Every data set has the same 87 variables available and has been randomly drawn from the total 411471 observations.

Using the Weight of evidence transformation, needs to be done carefully. If we would transform all data and then separate it into train, validation and test sets, the train set would contain information about the defaults in the entire population. We need to split the data first, transform the train data and then, using the bins of the train data WOE transformation, we transform the validation and test set.

Chapter 4

Machine Learning Retail Credit Risk models

In this chapter the test and validation data sets will be used to create models for the probability of charge off, exposure at charge off and loss given charge off. The expected loss will also be modeled directly besides modeling the separate quantities of which the experted loss consists. The algorithms that were discussed in Chapter 2 will be used with the train and validation data that was described in Chapter 3. Optimal models per quantity and algorithm are found by making models with the train data under different parameter settings, and selecting the model with best performance on the validation set. This approach prevents overfitting to the training data by looking at performance of new data.

4.1 Probability of charge off

Four approaches to estimate class probabilities of a loan belonging to the charged off loan status class will be discussed in the next sections, starting with traditional Logistic Regression and moving on to Random Forests, Neural Networks and Support Vector Machines.

The baseline that we set for the models is to predict the average charge off rate observed in the entire train set as a probability of charge off for the test observations. which is 19.149%. The AUC of this model is 0.5, this corresponds to having a model with zero predictive power.

4.1.1 Logistic Regression

The first step we take is creating a correlation matrix. This allows us to evaluate what variables show correlation to the variable that we aim to predict. Variables with a high positive or negative correlation to the target variable have high predictive power, when correlation is zero, the variable will, in general, not add predictive power to a regression model. The other maybe even more important reason for checking the correlation matrix is that a Logistic Regression model which is a linear model assumes independence between the model variables. When variables are correlated, assumptions and conclusions drawn from the fitted model might be misleading.

First we deal with finding a reasonable amount of variables to include in the model. By trying different target variable correlation thresholds. When the threshold is set at 0.07 we have 20 variables left from which we can fit models. All possible models with one, two, three, four and five variables are fitted to the data. The 21.699 models

are then compared on Akaike Information Criterion (AIC) (Hu, 2007).

The AIC measures how good a model is, relative to another model created in the same environment. Models are punished for having unnecessary complexity, which is done with the following formula:

$$AIC = 2k - 2ln(L)$$

where k represents the number of estimated parameters and L represents the maximized value of model likelihood. The value of the AIC itself has no meaning, but when we have models that are created in the same environment, we can select the best model by picking the one with the lowest AIC.

Model 1 summary:

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6710  -0.6811  -0.5370  -0.3555   2.9054

Coefficients:
.              Estimate   Std.Error z value  Pr(>|z|)
(Intercept)  -1.444171   0.005093 -283.58   <2e-16 ***
term          1.062204   0.011838   89.72   <2e-16 ***
APPL_FICO_BAND 0.939969   0.014545   64.63   <2e-16 ***
dti           0.673021   0.016208   41.52   <2e-16 ***
zip_code      0.851267   0.021584   39.44   <2e-16 ***
annual_inc    0.948192   0.024528   38.66   <2e-16 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 273493  on 279999  degrees of freedom
Residual deviance: 253824  on 279994  degrees of freedom
AIC: 253836

Number of Fisher Scoring iterations: 5

AUC: 0.6802947
```

The selected model has five significant variables, meaning that the coefficients of the variables in the model are significantly different from zero for which the hypothesis was tested.

Another approach to try, is to include all 20 previously selected variables in the model, and remove insignificant variables until until the model with lowest AIC is found. This method is called stepwise regression with backwards elimination.

Model 2 summary:

```
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.6425  -0.6818  -0.5272  -0.3404   2.9740

Coefficients:
                        Estimate   Std.Error  z value  Pr(>|z|)
(Intercept)           -1.444828   0.005126 -281.846  < 2e-16 ***
term                   1.028194   0.012296   83.617  < 2e-16 ***
APPL_FICO_BAND         0.702271   0.017536   40.047  < 2e-16 ***
dti                    0.475499   0.017889   26.580  < 2e-16 ***
bc_open_to_buy         0.814874   0.054250   15.021  < 2e-16 ***
percent_bc_gt_75       0.137116   0.039355    3.484 0.000494 ***
avg_cur_bal            0.389631   0.046667    8.349  < 2e-16 ***
zip_code               0.848744   0.021732   39.055  < 2e-16 ***
bc_util               -0.516950   0.050701  -10.196  < 2e-16 ***
tot_cur_bal           -0.147237   0.048624   -3.028 0.002461 **
annual_inc             0.941332   0.028286   33.279  < 2e-16 ***
verification_status    0.486283   0.024300   20.011  < 2e-16 ***
acc_open_past_24mths   0.504892   0.032761   15.412  < 2e-16 ***
revol_util             0.305937   0.036711    8.334  < 2e-16 ***
mort_acc               0.539045   0.032901   16.384  < 2e-16 ***
total_bc_limit        -0.689281   0.058266  -11.830  < 2e-16 ***
mo_sin_rcnt_tl         0.328387   0.036420    9.017  < 2e-16 ***
num_actv_rev_tl       -0.229393   0.037312   -6.148 7.85e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 273493  on 279999  degrees of freedom
Residual deviance: 251745  on 279982  degrees of freedom
AIC: 251781

Number of Fisher Scoring iterations: 5

AUC: 0.6910932
```

The second model has lower AIC and higher AUC, and is therefore, according to an econometric perspective, a preferable model. The models were fitted with 280,000 observations, and the AUC was calculated on a validation set of 120,000 observations. This model is sensitive to the amount of data used. When the model is fitted with 28,000 loans, the AUC on the validation set drops to 0.67 and four variable coefficients have become statistically insignificant for the model.

4.1.2 Random Forest

For Random Forest models, we will use the Ranger package in R. The name of the software package is derived from the words, RANdom forest GEneRator (Wright and Ziegler, 2015). This package is superior mainly in terms of speed compared to other Random Forest software. While we were concerned with the amount of input variables in other models, that aspect is less important with Random Forests. As explained in Section 2.3.2 a Random Forest can disregard unimportant variables by randomly drawing a higher amount of candidate split variables. Also training a Random Forest with all variables enables us to create a variable importance table indicating what variables are most important for growing the trees in the Random Forest. The variable importance in Table 4.1 shows the top ten most important variables for growing a probability of charge off Random Forest. The full variable importance table is stated in Appendix D.

variable	importance	variable	importance
zip_code	416.0527	annual_inc	154.1063
APPL_FICO_BAND	213.8836	issue_month	149.6464
addr_state	198.6785	emp_length	145.0916
term	189.9088	revol_util	135.0623
dti	182.1208	total_acc	129.1624

TABLE 4.1: Random Forest probability of charge off top ten most important variables

We can see that there is a clear winner among the available variables, the zip code. It is almost twice as important as the second most important variable. We will keep it in the model in order to achieve high performance, however from an ethical point of view the use of this variable is of course questionable.

A standard Random Forest creates five hundred trees, randomly selects the rounded square root of the available variables as candidates for the splits and sets a minimal node size of 10 in case of probability estimation. These settings and the amount of data used to train the model have to be optimized in terms of area under the receiver operating characteristic curve.

Adding more data to the model will, to a certain extent, improve the model but is also expensive in computing time. Therefore we start with a reasonable amount of data to get a sense of the behavior of the forest under different parameter settings. When we have found good parameters, we will improve performance by adding data. Ten thousand is a reasonable amount to create models, this corresponds to 4286 observations in the validation set in order to keep the 70/30 percent fractions in tact. The standard Random Forest model achieves a performance on the validation data of 0.65614 AUC. The performance on the training data is 0.99999 AUC. These AUC's correspond to the ROC plots in the figure below.

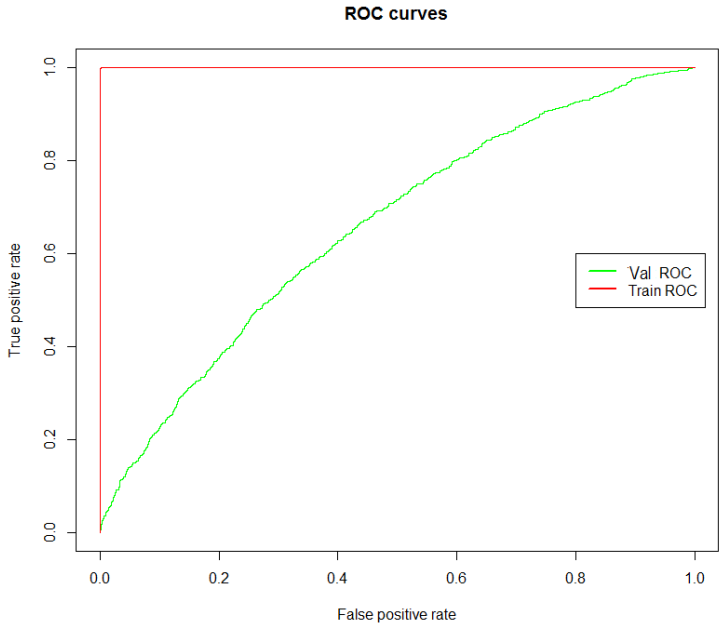


FIGURE 4.1: Random Foret initial ROC performance on train and validation data

To achieve higher performance on validation data we need to optimize the model to generalize better. To make the model less complex we can change the depth parameter or we can increase the number of variables randomly chosen at each split.

Now we have to test different values of the training set size, the amount of variables randomly selected for split criterion and the terminal node size and the number of trees to put in the forest.

Min node size	Train AUC	Validation AUC
10	1	0.66078
100	0.93406	0.66289
200	**0.85902**	**0.66321**
300	0.82801	0.66249
400	0.81170	0.66178
500	0.78902	0.66028
600	0.78392	0.65840
700	0.78825	0.65886
800	0.78502	0.65899
900	0.76259	0.65732
1000	0.76621	0.65655

TABLE 4.2: Random Forest Train and validation performance on different values of minimal node size

From Table 4.2 we conclude that 2% of train set size is a good setting for the depth parameter. The number of variables, to draw from all variables, as candidates to make a split on will be tested on a range, close to the default setting, the rounded square root of all variables.

mTry	Train AUC	Validation AUC
2	0.84027	0.67326
3	**0.85284**	**0.67431**
4	0.85745	0.67335
5	0.85947	0.67149
6	0.86009	0.66958
7	0.86010	0.66678
8	0.85966	0.66534
9	0.85946	0.66370
10	0.85924	0.66140
11	0.85821	0.65908
12	0.85751	0.65789
13	0.85720	0.65729

TABLE 4.3: Random Forest Train and validation performance on number of variables as split candidates (mTry)

In Table 4.3 we observe that lowering the amount of candidate variables increases the AUC on the validation set. We stated earlier that a higher value allows for a less complex forest and should reduce the difference between the performance on train and validation data. The higher performance with a lower mTry, can be explained by the fact that a high amount of candidates will will make the model choose the best variable, in terms of train data, to make the split. Having less freedom in picking the split variable makes the forest able to make better generalizations, which is reflected in the performance on validation data.

Trees	Train AUC	Validation AUC
100	0.85925	0.67108
200	0.83736	0.67279
300	0.84939	0.67342
400	0.85422	0.67436
500	0.84715	0.67459
600	0.85472	0.67432
700	**0.86040**	**0.67568**
800	0.86279	0.67477
900	0.84281	0.67471
1000	0.84847	0.67463

TABLE 4.4: Random Forest Train and validation performance on number of trees in the forest

Table 4.4 shows that the optimal size of the forest is 700 trees.

Final forest

Running the Random Forest on the entire train data set, with the parameters found on the smaller set, an AUC of 0.7129 is achieved. Adding data to the forest produces a substantial performance gain, because the larger set is a better representation of the validation data. By repeating the parameter search, that was performed with less data, on the larger data set, other parameters are found that produce an even higher AUC. When a Random Forest is trained on 280,000 observations with 1,000 trees, mTry 7 and node size 1,000 we achieve the highest AUC which is 0.7231. From this we can conclude that training a forest with more data needs re evaluation of the

parameters.

For comparison, we also run a Random Forest without the weight of evidence transformation, using the same model parameters, from re evaluation of the larger data set forest. This model achieves an area under the receiver operating characteristic of 0.7247 on the validation data.

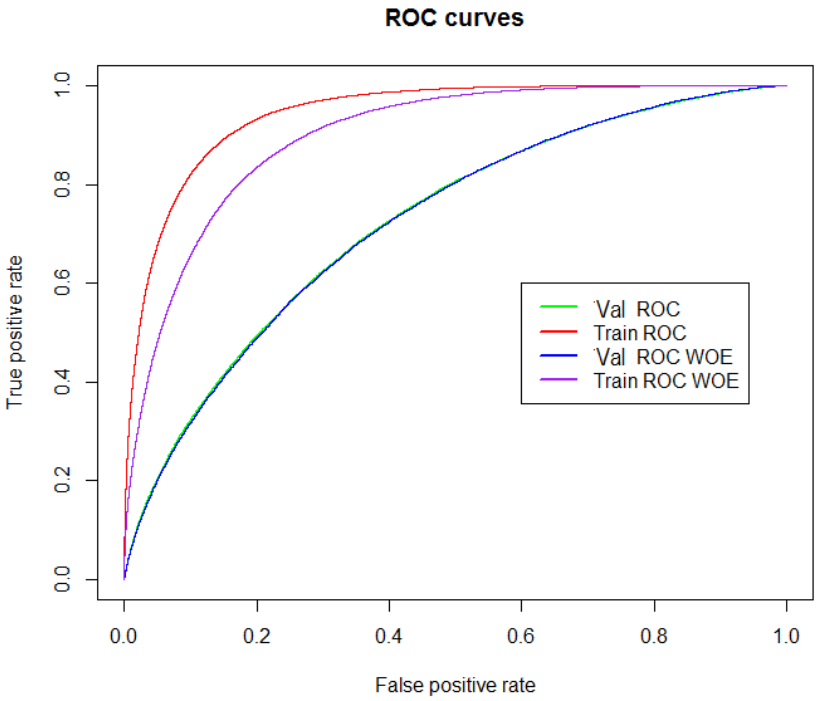


FIGURE 4.2: Random Forest ROC performance of final model using original and WOE transformed data

Figure 4.2 shows that on the validation set, the model using original data performs similar to the model using weight of evidence data. However the better performance on the train set might indicate that there is more room for improvement for the model using original data. The highest AUC achieved by the WOE based forest is 0.7232. The original data model achieved a higher AUC of 0.7244.

4.1.3 Neural Network

Stuttgart Neural Network Simulator software for R (RSNNS) is used for our analysis of Neural Network models. The parameters in this model that we have to optimize for our data set are training set size and number of neurons in the hidden layer.

When we include all features and increase the number of neurons in the hidden layer, we are gradually introducing more complexity into the model. With a small amount of data, an over fit will be produced quickly without adding much neurons in the hidden layer. This gradually adding of complexity will be done with 1, 10 and 100 percent of the data. The following plots and Table 4.5 represent the results of this process.

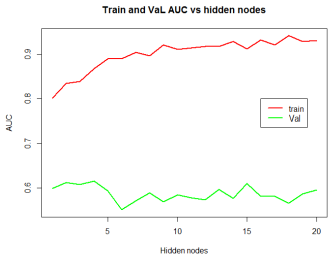


FIGURE 4.3: 1%Data

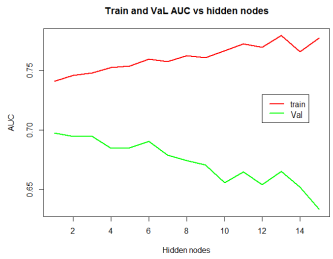


FIGURE 4.4: 10%Data

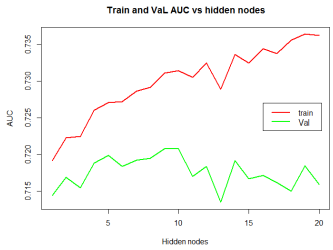


FIGURE 4.5: 100%Data

FIGURE 4.6: Neural Network Train and Validation AUC with different data set sizes and numbers of hidden nodes

Data	2800/1200		28000/12000		280000/120000	
nodes	AUC train	AUC Val	AUC train	AUC Val	AUC train	AUC Val
1	0.80102	0.59875	0.74096	0.69756	0.71916	0.71444
2	0.83398	0.61273	0.74603	0.69452	0.72227	0.71690
3	0.83925	0.60763	0.74782	0.69439	0.72250	0.71551
4	0.86734	0.61534	0.75262	0.68463	0.72605	0.71883
5	0.88974	0.59378	0.75375	0.68486	0.72708	0.71989
6	0.88953	0.55129	0.75977	0.69036	0.72722	0.71840
7	0.90359	0.57134	0.75739	0.67900	0.72863	0.71927
8	0.89578	0.58984	0.76244	0.67445	0.72917	0.71954
9	0.92039	0.56894	0.76077	0.67059	0.73108	0.72081
10	0.91013	0.58532	0.76672	0.65554	0.73146	0.72078
11	0.91424	0.57794	0.77265	0.66478	0.73050	0.71703
12	0.91745	0.57414	0.76937	0.65410	0.73247	0.71837
13	0.91761	0.59704	0.77937	0.66527	0.72890	0.71351
14	0.92848	0.57680	0.76568	0.65192	0.73365	0.71920
15	0.91170	0.61032	0.77747	0.63306	0.73246	0.71674

TABLE 4.5: Train and Validation AUC with different set sizes

The measure of complexity in the model needed for optimal performance on the validation set increases with the size of the data set. In Table 4.5 we see that the smallest train/validation set requires four hidden nodes achieving 0.62 AUC, with six hidden nodes an optimal validation AUC of 0.69 is achieved in the medium data set size. The Neural Network with 100 percent of the train data is able to achieve an AUC of 0.72 with 9 hidden nodes.

The probability of charge off Neural Network is very sensitive to the amount of data and the number of hidden nodes in the network. The best model is a Neural Network with 87 input nodes, 9 hidden nodes and one output node, resulting in a validation set AUC of 0.7208.

4.1.4 Support Vector Machines

Within the development of a Support Vector Machine suitable for predicting probability of charge off on the Lending Club data, we will try different settings on a

small data set. Optimizing on more data proves to be very time consuming, because the Support Vector Machine needs to perform calculations on every pair of observations, the amount of calculations grows very fast by adding data.

Support Vector Machines are said to be an excellent model choice when the data set suffers from the dimensionality curse (having a lot of variables and relatively few observations). Clearly with 87 variables and 400k observations, the Lending Club data set does not suffer from this curse. Leading us to expecting inferior behavior of Support Vector Machines, however they might be able to show good results when trained with a small amount of observations.

Kernel	Train AUC	Val AUC
Linear	0.5842	0.5938
Polynomial	0.9485	0.6508
Sigmoid	0.6003	0.6038
Radial	0.9514	0.6706

TABLE 4.6: Performance of Support Vector Machines trained with 2800 observations

The Support Vector Machine with the radial basis function kernel performs best on standard model settings, finding the optimal settings for this kernel is therefore most promising.

Gamma controls how much influence support vectors have, when gamma is low data points have a large influence region and this can lead to an under fit. High gamma on the other hand limits the region of influence of the support vectors and leads to over fitting to the training data. A gamma of 1/18 is found to be optimal for the validation data set.

Training a Support Vector Machine with radial kernel and gamma 1/18 on 28000 observations, results in 0.9947 and 0.6506 AUC on the train and validation data respectively.

Training the Support Vector Machine with 2800 observations takes 15 seconds, training the same model with 28000 observations costs 5644 seconds. The training time of an SVM with radial basis function kernel on 280,000 observations is estimated to be larger than 5644/15 = 376.27 times 1.5 hours, under the assumption that training time grows linear with the data set size. This assumption gives us a lower bound of roughly 564 hours, because in reality the training time grows much faster than linear with the data set size.

4.2 Exposure At Charge off

The next quantities that we model with the machine learning algorithms are continuous, they are modeled as a percentage of the total loan amount. The exposure at charge off can be derived from the Lending Club data with the following formula:

$$Exposure\ At\ Charge\ Off = 1 - \frac{total\ Principal\ Received}{total\ Loan\ Amount}$$

As baseline for this model, we take the average of the train data and predict this for the validation data. resulting in a mean squared error on the validation set of:

$$base\ MSE = 0.03955$$

The exposure at charge off must be modeled with data from charged off loans. The same initial split of 280000/120000 is made. After splitting the data, the charged off loans are selected resulting in 53618/22825 train/validation observations, which approximately still is a 70/30 percent split.

4.2.1 Regression

Using stepwise regression, with backwards elimination that minimizes the AIC, we find a model with 28 variables. intercorrelated variables were removed, the variable of the correlated pair with highest correlation to the response variable is kept in the initial regression model. The model mean squared error is 0.02778 with an R^2 of 0.29770, these values were calculated using the validation data. The model summary can be found in Appendix E.1.

4.2.2 Random Forest

From modeling the probability of charge off we know that parameters that show good performance with a small subset of the data, are not the best parameters when the forest is trained on a larger set. Therefore we do not start with a subset of the charged off loans, but use all available observations for modeling. The most important features for the Random Forest exposure at charge off model are shown in Table 4.7, note that the numbers are not comparable with the importance of the probability of charge off forest. These numbers are related to the variance in the nodes, the probability of charge off forest importance is related to the Gini in the nodes. The complete variable importance table for the exposure at charge off is available in Appendix D.

variable	importance	variable	importance
term	199.266	dti_joint	41.034
zip_code	65.897	dti	38.682
addr_state	52.330	emp_length	37.889
APPL_FICO_BAND	50.576	annual_inc	37.242
issue_month	47.069	revol_util	36.901

TABLE 4.7: Random Forest exposure at charge off top ten most important variables

Through a parameter search, presented in Table 4.8 we find the best parameters of the forest to be a random drawing of 20 variables to try as split variables for each split, nodes that contain more than 25 observations will be split and the forest of regression trees will contain 1200 individual trees.

mTry	Train MSE	Val MSE	nTree	Train MSE	Val MSE	Node Size	Train MSE	Val MSE
20	0.004904	0.026622	1200	0.006075	0.026822	25	0.015001	0.026787
28	0.004725	0.026636	1400	0.006095	0.026841	40	0.018066	0.026790
30	0.004697	0.026642	1000	0.006077	0.026848	44	0.018649	0.026800
22	0.004846	0.026648	1100	0.006094	0.026849	22	0.014108	0.026806
24	0.004798	0.026653	1500	0.006073	0.026851	27	0.015521	0.026809
29	0.004714	0.026654	900	0.006085	0.026851	31	0.016440	0.026811
26	0.004759	0.026656	1000	0.006082	0.026855	43	0.018500	0.026813
23	0.004821	0.026664	1300	0.006069	0.026865	29	0.015998	0.026815
18	0.004975	0.026669	800	0.006079	0.026865	41	0.018214	0.026816
21	0.004874	0.026670	600	0.006088	0.026877	33	0.016858	0.026818

TABLE 4.8: Random Forest EAC model parameter search sorted on validation performance

In the table there are parameters missing, the table only shows the top ten sorted on validation MSE. Table D.5 in the appendix gives a complete overview of performance on all parameters that were tested.

The exposure at charge off Random Forest model achieves 0.012972 train MSE and 0.026565 validation MSE. The R^2 on the validation set is 0.328295. A Random Forest with the unprocessed data results in a MSE on the validation set of 0.026509 with an R^2 of 0.329690.

4.2.3 Neural Network

We have noticed the same behavior with Neural Networks as with Random Forest in the performance on the validation data under different data set sizes. Again we will start with all available data and proceed to find the right amount of nodes to put in the hidden layer. Figure 4.7 and Table 4.9 show the performance of networks with different numbers of hidden nodes controlling the amount of complexity that the network can include in the model.

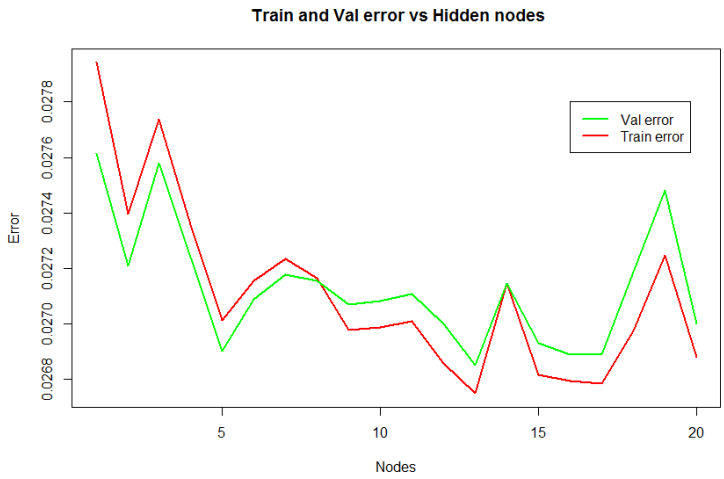


FIGURE 4.7: Neural Network EAC MSE on train and validation data vs number of hidden nodes

Nodes	Train MSE	Val MSE	Nodes	Train MSE	Val MSE
1	0.027943	0.027614	11	0.027008	0.027109
2	0.027397	0.027208	12	0.026857	0.026999
3	0.027738	0.027580	**13**	**0.026748**	**0.026850**
4	0.027359	0.027240	14	0.027145	0.027145
5	0.027013	0.026903	15	0.026815	0.026929
6	0.027156	0.027089	16	0.026794	0.026889
7	0.027234	0.027179	17	0.026784	0.026888
8	0.027166	0.027157	18	0.026973	0.027186
9	0.026979	0.027070	19	0.027247	0.027481
10	0.026988	0.027084	20	0.026878	0.026999

TABLE 4.9: Neural Network EAC train and validation performance

The Neural Network with 13 nodes in the hidden layer results in the lowest mean squared error on the validation set. The error is 0.02685 with an R^2 of 0.3188852.

4.2.4 Support Vector Machines

As opposed to the other EAC models, we have to search the best SVM parameters on a smaller set of charged off observations because searching parameters on the larger set would take too much time. For this purpose, 10% of the available data is used. The best performing kernel and parameter will then be applied to the large data set, creating the final model that we will later compare with other algorithms for modeling the exposure at charge off.

Kernel	Train MSE	Val MSE
Linear	0.02938	0.02720
Polynomial	0.02261	0.03053
Sigmoid	11.57833	10.51691
Radial	0.02191	0.02668

TABLE 4.10: Support Vector Machine EAC performance using different kernels

The results in Table 4.10 come from standard parameters of the models. In case of the polynomial kernel (degree 2), increasing or decreasing the degree makes performance of the model worse. The linear kernel has no parameter to improve and the sigmoid kernel is showing very bad performance on this data set. We continue with the most promising model, the Support Vector Machine with the radial basis function kernel. The best gamma parameter in terms of MSE on the validation data set is $\frac{1}{350}$.

On the large data set, the RBF Support Vector Machine achieves 0.02694668 MSE, 0.0271566 validation MSE and an R^2 of 0.3133397 on the validation set.

4.3 Loss Given Charge off

The baseline for this quantity is a mean squared error of 0.00870. This represents the error obtained when predicting the average of the train set for every validation set observation, and calculating the errors.

$$recovery\ rate = \frac{recoveries - collection\ recovery\ fee}{exposure}$$

$$loss\ given\ charge\ off = 1 - recovery\ rate$$

$$base\ MSE = 0.00870$$

For modeling the loss given charge off we use only charged off loans like we did in the previous section.

4.3.1 Regression

From the 87 variables we keep 37 that are not highly correlated to each other. The variables that were thrown out were less correlated to the response variable than the variables that are kept in the model. From the 37 variables we remove the insignificant ones one by one minimizing the AIC. The model with the lowest AIC has 25 variables included, the model summary is added in Appendix E.2 This results in an MSE of 0.008262 and and R^2 of 0.049975 on the validation set.

4.3.2 Random Forest

The same approach as for the Exposure at charge off Random Forest is taken for the Loss Given Charge Off Random Forest. The response variable is now the proportion off exposure that is not recovered. The variables in Table 4.11 are the variables that contribute the most to reducing variance inside the nodes, the complete table is stated in Appendix D.

variable	**importance**	**variable**	**importance**
zip_code	18.49764	dti	10.23981
addr_state	15.15434	annual_inc	9.981299
APPL_FICO_BAND	14.62438	total_acc	9.893213
issue_month	11.66216	revol_util	9.749394
emp_length	10.38577	revol_bal	9.078406

TABLE 4.11: Random Forest Loss Given Charge off top ten most important variables

In Table 4.12 the top parameters of a parameter search are presented. The entire parameter search is specified in Table D.6 in the appendix.

mTry	Train MSE	Val MSE	nTree	Train MSE	Val MSE	Node Size	Train MSE	Val MSE
9	0.002287	0.008203	1400	0.002290	0.008198	49	0.006367	0.008176
8	0.002406	0.008203	1200	0.002284	0.008199	47	0.006301	0.008177
11	0.002110	0.008208	600	0.002301	0.008200	48	0.006333	0.008177
10	0.002192	0.008209	1100	0.002289	0.008201	37	0.005884	0.008179
5	0.003042	0.008209	1300	0.002286	0.008201	44	0.006185	0.008179
7	0.002597	0.008209	900	0.002298	0.008202	42	0.006105	0.008180
14	0.001975	0.008210	1500	0.002289	0.008202	50	0.006401	0.008180
6	0.002802	0.008211	1000	0.002283	0.008206	39	0.005973	0.008180
13	0.001997	0.008212	800	0.002292	0.008206	36	0.005820	0.008180
3	0.004258	0.008215	500	0.002296	0.008207	41	0.006071	0.008181

TABLE 4.12: Random Forest Loss Given Charge off model parameter search sorted on validation MSE performance

A Random Forest with minimal node size 49, mtry 9 and 1400 trees is chosen as final model. This model achieves 0.00817 validation MSE with an R^2 of 0.05989. When we use the unprocessed data, the Random Forest achieves 0.00817 validation MSE with an R^2 of 0.06015.

4.3.3 Neural Network

The Loss Given Charge Off Network is unable to find a fit. There is a model that we can evaluate, however by looking at Figure 4.8, we see that the error on the validation set is always lower than the error on the train set. This can be interpreted as the validation observations being closer to their mean than the train observations. In that case a model that does almost nothing, which is confirmed by the low R^2, can perform better on the test set than on the train set.

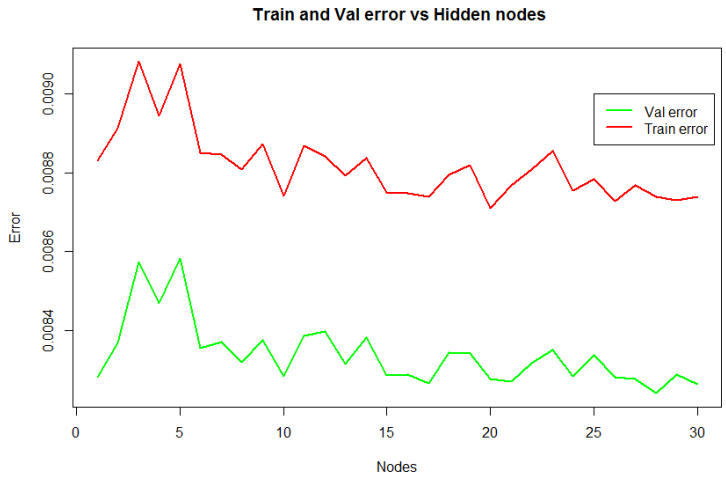


FIGURE 4.8: Neural Network LGC MSE on train and validation data vs number of hidden nodes

Nodes	Train MSE	Test MSE	Nodes	Train MSE	Test MSE
1	0.008831	0.008282	16	0.008747	0.008288
2	0.008913	0.008369	17	0.008738	0.008266
3	0.009082	0.008574	18	0.008795	0.008343
4	0.008943	0.008470	19	0.008819	0.008342
5	0.009075	0.008584	20	0.008711	0.008277
6	0.008850	0.008356	21	0.008768	0.008271
7	0.008845	0.008371	22	0.008807	0.008317
8	0.008809	0.008319	23	0.008854	0.008350
9	0.008872	0.008376	24	0.008755	0.008284
10	0.008742	0.008283	25	0.008783	0.008337
11	0.008869	0.008387	26	0.008728	0.008281
12	0.008841	0.008398	27	0.008767	0.008277
13	0.008793	0.008315	**28**	0.008738	**0.008240**
14	0.008837	0.008382	29	0.008729	0.008288
15	0.008751	0.008285	30	0.008739	0.008265

TABLE 4.13: Neural Network LGC train and validation performance

28 nodes in the hidden layer of the network gives the best performance. A MSE of 0.00824 and R^2 of 0.04550 are achieved with the single layer Neural Network.

4.3.4 Support Vector Machines

Table 4.14 is created by training and validating Support Vector Machines with different kernels on 10% of the available train and validation data sets.

Kernel	Train MSE	Val MSE
Linear	0.00923	0.00900
Polynomial	0.00657	0.00927
Sigmoid	2.51055	2.27503
Radial	0.00713	0.00892

TABLE 4.14: Support Vector Machine LGC performance using different kernels

The values in the table are calculated with the standard parameters of the models. In case of the polynomial kernel, decreasing the degree to two, makes performance of the model slightly better. However this is still not close to the performance of the radial basis function kernel.

The best gamma for the RBF kernel is $\frac{1}{600}$, a gamma that is this low leads to high influence regions of the potential support vectors. This would in general result in an under fit, however it is the best parameter on the validation data. Running this model on the large data set results in a validation MSE 0.00898 and R^2 -0.03267. This model shows good performance in terms of MSE, yet the negative R^2 contradicts good performance. A negative R^2 means that this model does worse that the baseline average prediction, which has an MSE of 0.00870.

4.4 Expected Loss

By dropping the assumption that PD EAD LGD or in our case PC, EAC and LGC are independent, it is no longer acceptable to just multiply these quantities in order to come up with the expected loss. In this section we will use the algorithms predict the expected loss of loans directly. Because we do not condition on the status of a loan, which was done with EAC and LGC, we use all available data for the expected loss models.

The average loss of principal across the train data is 13.219%, if we predict this for all validation observations, the mean squared error that we find is 0.08149.

4.4.1 Regression

After removing correlated variables, a stepwise regression model with backwards elimination is created. The resulting model, summarized in Table E.3, with lowest AIC consists of 33 independent variables and performance on the validation data in terms of MSE is 0.07276 and the R^2 is 0.10712.

4.4.2 Random Forest

The expected loss Random Forest, identifies variables in Table 4.15 as most important in the tree growing process. The complete variable importance table for the Random Forest expected loss model is available in Appendix D.

variable	importance	variable	importance
zip_code	916.0333	issue_month	501.9930
term	812.2799	annual_inc	484.9612
APPL_FICO_BAND	683.3494	emp_length	465.9498
addr_state	637.8747	revol_util	433.4505
dti	570.3449	total_acc	426.7132

TABLE 4.15: Random Forest Expected Loss top ten most important variables

Because the size of the train set is larger, we take steps of 10 in the search for the best minimal node size. Memory limitations prevented the training of forests larger than 900 trees.

mTry	Train MSE	Val MSE	nTree	Train MSE	Val MSE	Node Size	Train MSE	Val MSE
8	0.017405	0.071639	800	0.016767	0.071558	40	0.047228	0.071421
12	0.015657	0.071672	700	0.016786	0.071606	60	0.052848	0.071429
11	0.015946	0.071673	900	0.016749	0.071618	50	0.050439	0.071432
10	0.016312	0.071674	500	0.016789	0.071632	80	0.056276	0.071457
9	0.016790	0.071681	600	0.016771	0.071650	70	0.054765	0.071470
13	0.015410	0.071683	400	0.016831	0.071712	30	0.042813	0.071472
7	0.018226	0.071705	300	0.016831	0.071720	90	0.057576	0.071481
14	0.015239	0.071718	200	0.016878	0.071881	20	0.036165	0.071483
6	0.019430	0.071727	100	0.017119	0.072140	100	0.058661	0.071497
15	0.015056	0.071764				10	0.025125	0.071579

TABLE 4.16: Random Forest Expected Loss model parameter search sorted on validation MSE performance

The best performing parameters, shown in Table 4.16, are a minimal node size of 40, mTry 8 and 800 trees. This Random Forest achieves 0.071451 validation MSE and an R^2 of 0.123163. Using the original data a Random Forest with the same parameters obtains a MSE of 0.071419 and an R^2 of 0.123556.

4.4.3 Neural Network

For Expected loss, the Neural Network model shows normal behavior in Figure 4.9. The train error is lower than the validation error, and by gradually adding complexity in the model, we first see an under fit, then the minimum of the validation error and after that more divergence between the validation and train error.

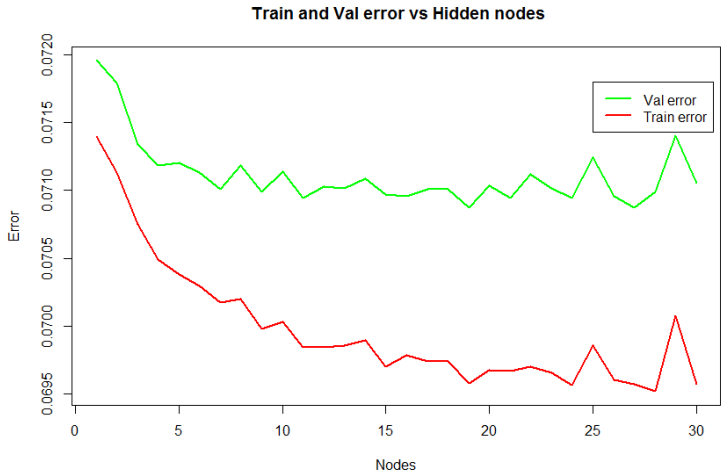


FIGURE 4.9: Neural Network EL MSE on train and validation data vs number of hidden nodes

Nodes	Train MSE	Val MSE	Nodes	Train MSE	Val MSE
1	0.071397	0.071959	16	0.069788	0.070959
2	0.071133	0.071794	17	0.069737	0.071009
3	0.070748	0.071341	18	0.069742	0.071008
4	0.070489	0.071187	19	0.069582	0.070872
5	0.070384	0.071204	20	0.069674	0.071034
6	0.070296	0.071132	21	0.069667	0.070947
7	0.070175	0.071010	22	0.069703	0.071117
8	0.070202	0.071183	23	0.069656	0.071012
9	0.069980	0.070991	24	0.069569	0.070945
10	0.070030	0.071140	25	0.069857	0.071243
11	0.069844	0.070947	26	0.069604	0.070958
12	0.069846	0.071031	**27**	0.069574	**0.070870**
13	0.069855	0.071017	28	0.069517	0.070989
14	0.069898	0.071084	29	0.070074	0.071402
15	0.069701	0.070968	30	0.069569	0.071056

TABLE 4.17: Neural Network EL train and validation performance

The best Neural Network for expected loss is a network with 27 nodes in the hidden layer of the network. This model achieves 0.070939 validation MSE and an R^2 of 0.129425.

4.4.4 Support Vector Machines

From modeling the probability of charge off we know that the amount of data is too large to handle for an SVM with the available resources. We also know that the radial kernel performs best for all other quantities modeled with SVM's. When we apply this kernel on a sample of 1% of the data set, we find an optimal gamma of $\frac{1}{16}$. The final model is trained with 10% of the data set. The model predictions on the validation data set have a mean squared error of 0.078713 and an R^2 of 0.019796.

Chapter 5

Model Analysis

Chapter 5 will evaluate and compare both the model approach and the model performance of (logistic) regression and the machine learning algorithms Random Forests, Neural Networks and Support Vector Machines. The model approach differences are addressed first, this will contribute to answering subquestion c. The next and also last subquestion about added value of machine learning in terms of model performance will be analyzed on prediction accuracy and on calibration

5.1 Modeling approach

The differences in approach between the regression and machine learning models that we have experienced in this research will be discussed in this section. Because the focus of this research has been to compare models in a fair way by using the same transformation, we can only evaluate what had to be done different after gathering data, selecting data and transforming the data.

- **(Logistic) Regression**
  During the modeling of probability of charge off with Logistic Regression and the continuous quantities with regression, we had to carefully select variables to include in the models. With the main concern that correlated variables produce unreliable models due to the independence assumption of regression models. The next step was to choose from the correlated variables which one to keep in the model. trying all possible combinations and subsets of combinations of uncorrelated variables was not possible because of the number of available variables. Our approach was to keep the variables with the highest correlation to the quantity that the model should predict. This step has led to a loss of potentially informative variables. Our following step was to use the AIC in order to select the best model from all created regression models. This is an often used statistical method to select the best regression model, it can easily be misused when it is used to compare models that are created in different data environments. When the AIC is used in a correct manner, it shows which model has the best trade off between complexity and performance. The model with the lowest AIC value is then selected as the final model.

- **Machine learning algorithms**
  The approach used with the machine learning algorithms was to first experiment with a relatively small amount of data, before training the algorithms on large amounts of data. This was needed, because training machine learning algorithms on large amounts of data can be a time consuming process. After taking this approach with modeling the probability of charge off, we found

that parameters that work well on a small data set are not the best parameters when more data is used. For the Support Vector Machines however we still had to apply the initial approach, because finding optimal parameters on large data sets is too time consuming. Parameters for the other models created during this research were searched using the largest available data sets. The optimal parameters are the parameters that perform best when the models make predictions on new data. we have seen in Chapter 4 that machine learning algorithms are in most cases able to create models that fit very good to the training data set but that the same model can perform very poor on new data. When the parameters that produce optimal out of sample results are found, we are finished and have a final model.

The machine learning approach saves time in the early stages of modeling, because there is no independence assumption that requires carefully disregarding parts of the data set. During the selection of the best parameters for a machine learning model, a lot of time is consumed when large data sets are used to train the models. Stepwise regression is a faster approach in finding the optimal model. On the other hand, a thorough approach in finding the optimal regression model would be to try all possible combinations of model variables. This becomes problematic with the amount of variables used in this research. Another difference between regression and machine learning is that the model selection based on AIC is using the model likelihood on training data. The machine learning models are selected by looking at performance on data that was not used during the training of the model.

A big advantage of the Random Forest algorithm is that it did not require the transformation of data to a scaled and numeric data set. The algorithm is even insensitive to outliers present in the data, where the regression and other machine learning approaches can suffer from the presence of such outliers in data sets. Another strong point of the Random Forest models is the ability to show what variables are the most important during the training of the Random Forest. When scaled data is used, regression model coefficients also show what variable contributes the most to the prediction of the model. Neural Networks and Support Vector Machines do not have such simple and intuitive ways of showing what variables are important for the credit risk predictions.

5.2 Model performance

For performance analysis, we have kept 11471 observations locked away, that we will now use to compare the different models. The validation data used in the previous chapter has been used to choose the best parameters, and is not suitable for comparing and analyzing the models. The chosen model parameters are optimal for the validation data and the models might have bias towards the particularities of this data. In this section, performance of the models will be evaluated on data that has not been used in any stage during the creation of the models.

5.2.1 Probability of Charge Off model

The probability of charge off models will be compared on ROC performance and we will compare how good the models are at assigning high probabilities of charge off to actually charged off loans. Figure 5.1 and Table 5.1 show the receiver operating characteristic curves and the area under that curve respectively.

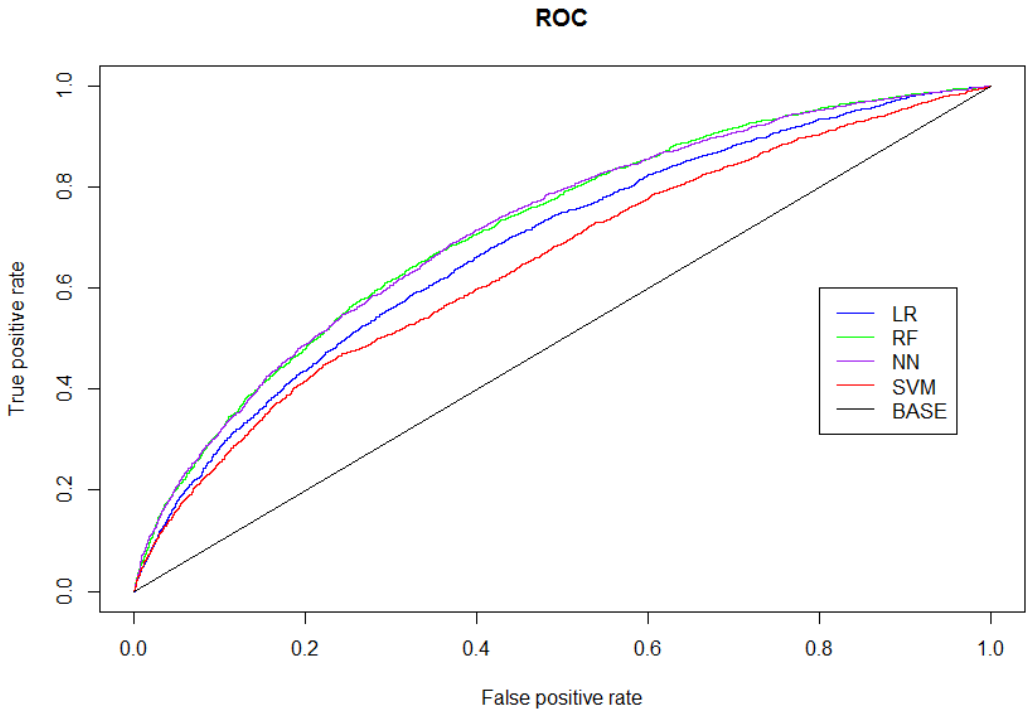


FIGURE 5.1: Probability of Charge off model ROC performance on test data

Model	AUC
Baseline	0.5
Logistic Regression	0.68366
Random Forest	**0.71782**
Neural Network	0.71747
Support Vector Machine*	0.64942

TABLE 5.1: Probability of Charge off model performance

First of all we notice that the Random Forest and the Neural Network clearly outperform the other models, secondly these two models are very close both in terms of ROC curve shape and AUC value. From these two models the probability that the Random Forest has assigned a higher probability of charge off to an observation of the charged off population when compared to an observation of the fully paid population is 0.035% higher.

The Random Forest on original data achieves 0.72053 AUC on the test data, with the settings that were optimal for the Weight of Evidence forest on the validation data. In Section 4.1.2 we have seen that the gap between train and validation ROC is larger for not transformed data than for the transformed data. This performance loss comes from restricting the Random Forest in its own decision making through binned data.

*Due to the computational expensiveness, the Support Vector Machine has not been run with the same data set size as was used with the other models.

When the probability of charge off predictions for 11471 test observations of all models are ordered on predicted probability of charge off, we can evaluate how much of the safe and risky loans, according to the models, were actually a good or bad investment. Tables 5.2 and 5.3 show the number of charged off loans in the thousand riskiest and safest loans respectively.

model	amount	percentage
LR	482	42.023%
RF	530	46.207%
NN	**537**	**46.818%**
SVM	461	40.192%

TABLE 5.2: Number of charged off loans in the 10% riskiest predicted loans

model	amount	percentage
LR	65	5.667%
RF	**50**	**4.359%**
NN	52	4.534%
SVM	110	9.590%

TABLE 5.3: Number of charged off loans in the 10% safest predicted loans

These tables show in a more intuitive way how good models are at concentrating possibly bad loans in the high probabilities of charge off and possibly good loans in the low probabilities of charge off. We already know that the Random Forest and Neural Network out perform Logistic Regression and the Support Vector Machine. These tables show however, that a Neural Network is slightly better at concentrating actually charged off loans in the high probability region and that a Random Forest is slightly better at concentrating loans that were fully repaid in the low probability of charge off region.

5.2.2 EAC and LGC models

The exposure and loss given charge off are modeled as a percentage of the total loan amount. Their performance on the test data set is evaluated by comparing mean square error and R^2 values.

Exposure At Charge off

Table 5.4 summarizes the performance results of the models that predict the exposure at charge ff of a loan.

Model	MSE	R^2
Baseline	0.03894	
Regression	0.02746	0.29447
Random Forest	**0.02644**	**0.32084**
Neural Network	0.02666	0.31522
Support Vector Machine	0.02674	0.31312

TABLE 5.4: EAC model performance

The mean squared errors for the machine learning models are all lower than the mean squared error of the regression model. This is also reflected in the R^2 that the models achieve. The lowest mean square error and highest R^2 is achieved by the Random Forest model. When the data that was not transformed is used the Random Forest model achieves 0.02628 MSE and an R^2 of 0.32485.

Loss Given Charge off

Table 5.5 summarizes the performance results of the models that predict the loss given charge off of a loan.

Model	MSE	R^2
Baseline	0.00980	
Regression	0.00933	0.04751
Random Forest	**0.00931**	**0.04895**
Neural Network	0.00968	0.01183
Support Vector Machine	0.01028	-0.04989

TABLE 5.5: LGC model performance

The models perform slightly better than the baseline average prediction, except for Support Vector Machine that performs worse than the average prediction and is able to achieve a negative R^2. The Random Forest again performs best, however in this case the runner up is not the Neural Network but regression. The forest with original data achieved a mean squared error of 0.00927 and an R^2 of 0.05258.

From the R^2 squared values we can conclude that the models for this quantity hardly have any predictive power because the amount of variance in the test data that can be explained by the models is very small. Recall that in Chapter 3 Table 3.5 shows that across all Lending Club sub grades the loss given charge off's are between 91.44% and 94.19%. The expectation that a predictive model would have very little power is confirmed by the low R^2 values and mean square errors close to the baseline mean square error which is small in absolute sense.

Weight of Evidence use in EAC LGC models

The used method transforms data to represent the log odds ratio of charge off. Valuable information may be lost because the model response variable is different from the response used in the transformation. However the variables that contain information about the log odds ratio of charged off loans and fully repaid loans are valuable, to some degree, for predicting other responses. When we examine the differences in performance of the WOE and original data Random Forests, we see that there are no huge differences and they are consistent with the minor performance

gain of the probability of charge off Random Forests that predict the same response as used for the Weight of Evidence transformation.

5.2.3 Expected Loss models

Two approaches for predicting the expected loss will be evaluated, the first is where we multiply the individual components of expected loss and the second is where we predicted the losses to be expected at once, modeling observed loss as a response value.

For every algorithm we have multiplied the predictions of its probability of charge off, exposure at charge off and loss given charge off. Together with the observed losses the model performance is evaluated, which is represented in Table 5.6.

Model	MSE	R^2
Baseline	0.08145	
(Logistic) Regression	0.07351	0.09751
Random Forest	0.07151	0.12200
Neural Network	**0.07119**	**0.12601**
Support Vector Machine	0.07527	0.07589

TABLE 5.6: EL from separate PC, EAC and LGC model predictions

When the models are used to predict the expected losses without first modeling the separate components of expected loss, the performance reported in Table 5.7 is achieved.

Model	MSE	R^2
Baseline	0.08145	
Regression	0.07279	0.10626
Random Forest	0.07147	0.12255
Neural Network	**0.07029**	**0.13697**
Support Vector Machine	0.07929	0.02648

TABLE 5.7: Individual EL model performance

If we compare the two expected loss approaches, we see that except for Support Vector Machines, the models perform better when expected loss is modeled directly. The reason that Support Vector Machines perform better in the separate model setting is that the support vector models for exposure at charge off and loss given charge off were trained with the same amount of data as the other exposure and charge off models. The Support Vector Machine could not be trained with all data for the probability of charge off and the expected loss models. So for the separate model case the SVM is in disadvantage only with modeling the probability of charge off. For the expected loss in one model the Support Vector Machine is in disadvantage for the entire loss prediction. In both approaches the Neural Network models outperform the other models and the Random Forest comes close to the Neural Network performance.

Loss capture evaluation

To evaluate how good the models can capture losses, the expected losses predicted by the models are sorted and plotted against the observed losses. Figure 5.2 shows

the ability to capture losses of the separate model EL approach and Figure 5.3 shows the same for the individual EL models.

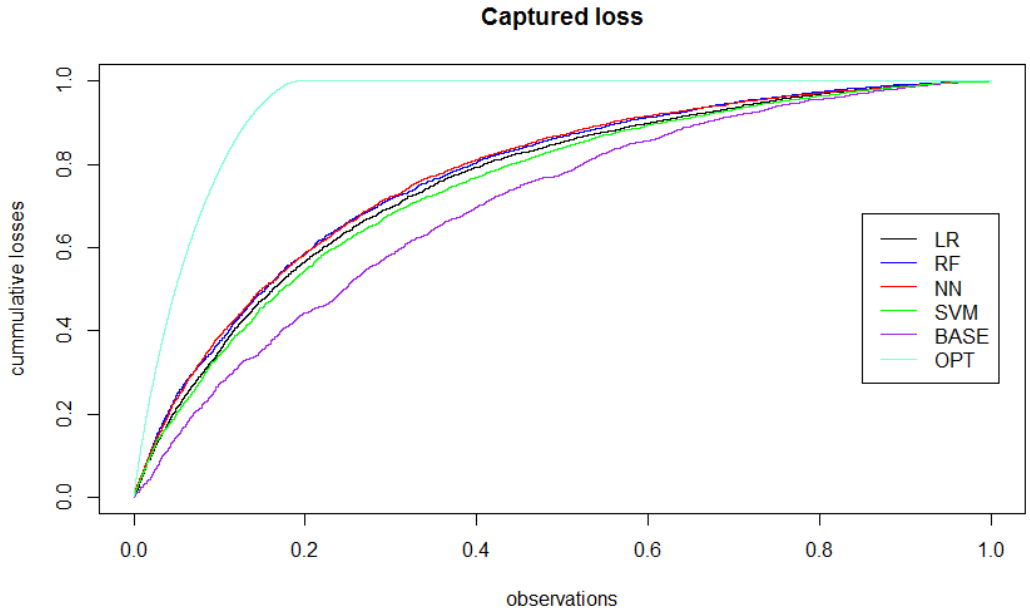


FIGURE 5.2: Loss capture plot separate models

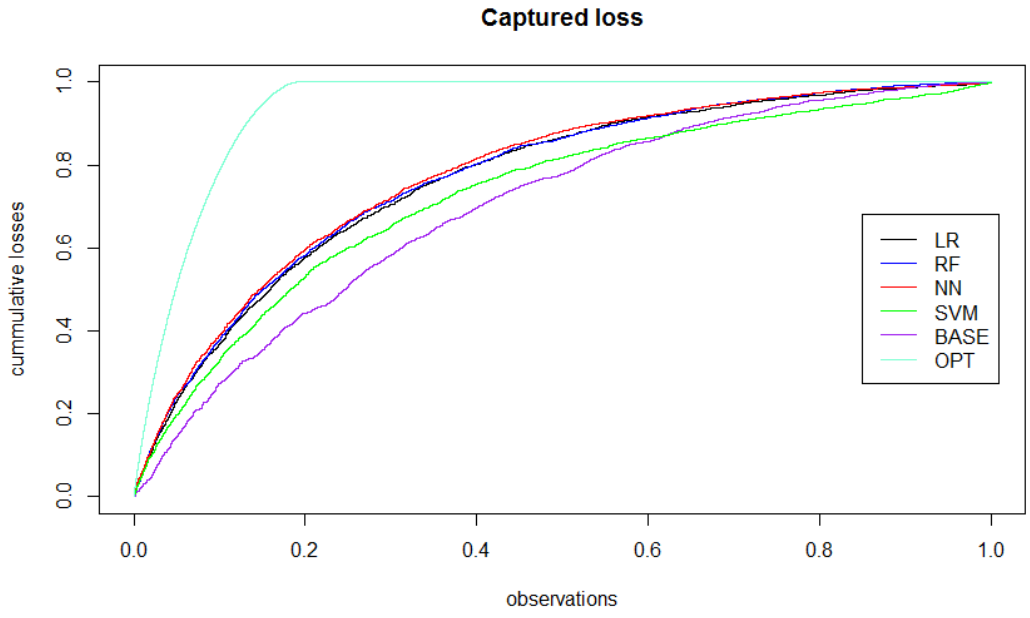


FIGURE 5.3: Loss capture plot individual models

The areas under the curves in figures 5.2 and 5.3 indicate how good a model can capture losses, an area of 0.5 indicates that a model does nothing. An optimal model (OPT in the figures), ordering all the losses from largest to smallest, would look like the aquamarine line. This line shows that roughly 19% of the loans in the test set

have been charged off and resulted in a loss, after that point the loss captured by the perfect model is 100% and it becomes a straight line parallel to the x axis.

Model	Separate	Individual
Baseline	0.70088	0.70088
Optimal model	0.94033	0.94033
Regression	0.76078	0.76890
Random Forest	0.77345	0.77330
Neural Network	**0.77506**	**0.77884**
Support Vector Machine	0.74969	0.72859

TABLE 5.8: Loss capture AUC

The Neural Network performs best in concentrating the actual losses in the loans that have higher expected losses. The single model approach slightly outperforms the expected loss performance from creating separate models for the probability of charge off, exposure at charge off and loss given charge off. In case of the regression model, the individual model also slightly outperforms the loss capture abilities of the separate model approaches. In case of the Random Forests and Support Vector Machines the predictions form separate components are better in capturing losses with high expected loss predictions. The baseline model in figures 5.2 and 5.3 represents a ranking of the loans on loan amount, it is able to achieve an area under the loss capture curve a lot higher than 0.5. This means that the loan amount can be seen as an important risk driver in the Lending Club data.

Calibration

In the previous sections we have analyzed how accurate the models are at individual loan level and how good the models are at assigning a high loss expectation to loans that have actually ben charged off. In this section we will let the models put the loans in five buckets ranging from highest expected loss (bucket 1) to lowest expected loss (bucket 5). In these buckets the total predicted loss is compared with the actual loss of all loans in the bucket. The figures 5.4 and 5.5 show the difference between the two as a percentage of the actual loss in the bucket, a positive bar in the figures indicate that the model expected more loss than what was actually observed and a negative bar indicates the opposite. When bars are small and close to zero, the model has predicted a loss in the bucket close to the actual sum of losses observed in the bucket.

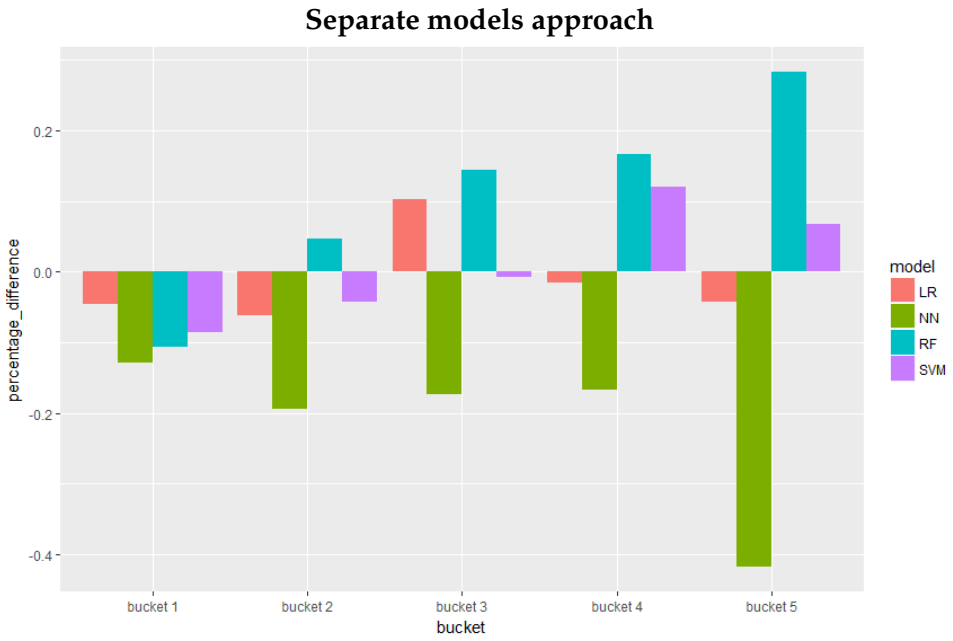


FIGURE 5.4: Percentage difference between predicted and actual losses in risk buckets of expected loss through separate models approach

In the separate model approach, Logistic Regression combined with standard regression and the Support Vector Machine models predict bucket losses closest to their actual values. Furthermore Neural Network models underestimate the losses in all buckets and the Random Forest models underestimate the losses in the most risky bucket while overestimating the losses in the other buckets.

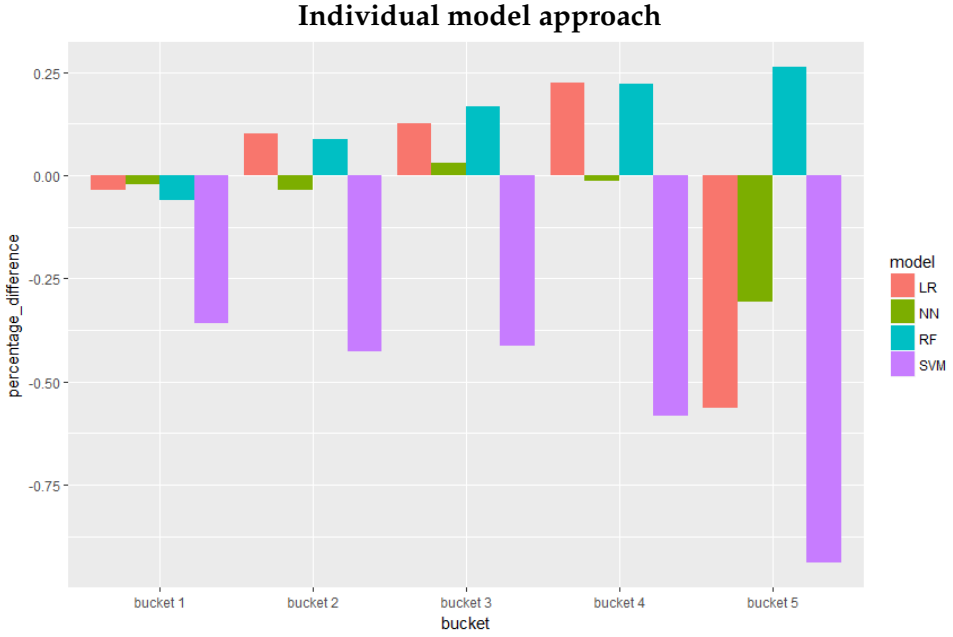


FIGURE 5.5: Percentage difference between predicted and actual losses in risk buckets of expected loss through individual model approach

The more direct single model approach for expected loss has enabled the Neural

Network to make better calibrated expected loss predictions. The Random Forest shows the same pattern of under and over estimation of losses as the separate model Random Forests, with a less severe underestimation of the losses in the most risky bucket. What we can also see is that the Regression approach is not able to make better calibrated predictions when the expected loss is directly modeled and that the Support Vector Machine model has very poor calibration also note that the predicted loss of the least risky bucket is more than 75% less than the actual losses of the loans in that bucket.

Chapter 6

Conclusions and Recommendations

This research investigated two possible sources of the added value of machine learning in retail credit risk. The first source being the modeling approach of traditional credit risk modeling versus the approach of machine learning algorithms. The second source of possible added value was evaluated by comparing model performance. After the conclusions about added value of machine learning in retail credit risk are discussed, the recommendations regarding future research will be discussed.

Conclusions

- *Added value in modeling approach*
  The added value of machine learning in terms of modeling approach is that in retail credit risk it can cope with an important development, the availability of large amounts of information on loan applicants. When there are more features then we can evaluate, it becomes hard to find the most suitable subset for a traditional retail credit risk model and to take care of correlated variables. The strong point of machine learning is that the algorithms used in this research can handle a very large amount of variables and that we can let the algorithms decide what features are important and how the combinations of these should be translated into predictions. Other added value that should be taken from the machine learning approach is that model selection based on data that was not used during the training or fitting stage puts the focus on what is actually important, the prediction quality on new observations. The traditional retail credit risk approach used in this research selects the best model with the Akaike Information Criterion, this measure does not evaluate the performance on new data.
  One specific model can add a lot of value in modeling approach. The Random Forest model does not need transformed or scaled data and is not sensitive to outliers. This can save a lot of time in the process of creating a retail credit risk model. The Random Forest algorithm can also be used to identify what features are important for risk prediction, the identified features could then be used in other models.

- *Added value in model performance*
  We have shown that the added value in retail credit risk, of machine learning over traditional credit scoring, in terms of performance is present. We have first created a level playing field by making the same information available to all models by means of Weight of Evidence transformation, and putting the same observations in the train, validation and test data. Of the investigated models both Random Forests and Neural Networks show better performance

than (logistic) regression. The two algorithms outscore Logistic Regression in terms of ROC AUC for probability of charge off prediction. The Neural Networks and Random forests have lower prediction errors and can account for a higher amount of variance in the predicted variable on the continuous quantities exposure at charge off and the expected loss in one model. For the loss given charge off predictions, none of the models show good performance because variance in that variable is already very low. The Neural Network was unable to find a fit and did not perform better than the regression model. The Support Vector Machine even achieved a negative R^2 on loss given charge off prediction, the Random Forest however performed slightly better than the regression model. When we look at calibration instead of accuracy, we see that the separate Logistic Regression and regression models combined into an expected loss prediction show better performance than the machine learning models. However the best calibrated model is the Neural Network that directly models the expected loss. This model is also the most accurate of the machine learning models. The Support Vector Machine models were not able to show good results because the data set was too large and the models created with smaller data sets could not come close to the performance of the other models.

To put these performance results into perspective we have to address the fact that the Weight of Evidence data transformation has not been beneficial for the Random Forests and possibly also for the other machine learning algorithms. Because the Random Forests did not require a data transformation we were able to also show the results of the model with original data in Chapter 4. These results show that the Random Forests without transformed data consistently outperform the WOE Random Forests. We suspect that the main reason for this is that binning the numerical data has given the models less freedom in making their own decisions.
The Weight of Evidence method was used to create a level playing field, in Chapter 2 we discussed why this method was most suitable for Logistic Regression, yet Logistic Regression is outperformed on WOE transformed data by the Random Forests and Neural Networks which strengthens our conclusion about the presence of added value in terms of performance in retail credit risk.

Recommendations for future reseach

- The next step in machine learning research on retail credit risk data such as the Lending Club data would be to evaluate the added value of online learning algorithms. These algorithms are constantly updated when new data becomes available. This would be valuable because when models adjust themselves, there is no need to invest in creating new models when a lot of new data is available or macro economic circumstances have changed.

- In this reseach we have used Neural Networks with one hidden layer. When more hidden layers are added the algorithm is called Deep Learning. It will be interesting to see if this method can perform better because of its ability to create more complex models.

- We have used the Weight of Evidence transformation in combination with Logistic Regression, Random Forests, Neural Networks and Support Vector Machines. A possible direction for further research is to evaluate the impact on

performance when data transformations with other link functions such as the probit instead of the logit from Weight of Evidence are used.

Appendix A

Variable description

Name	Description
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
acc_open_past_24mths	Number of trades opened in past 24 months.
addr_state	The state provided by the borrower in the loan application
all_util	Balance to credit limit on all trades
annual_inc	The self-reported annual income provided by the borrower during registration.
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
avg_cur_bal	Average current balance of all accounts
bc_open_to_buy	Total open to buy on revolving bankcards.
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
chargeoff_within_12_mths	Number of charge-offs within 12 months
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
desc	Loan description provided by the borrower
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income

Name	Description
earliest_cr_line	The month the borrower's earliest reported credit line was opened
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
emp_title	The job title supplied by the Borrower when applying for the loan.
fico_range_high	The upper boundary range the borrower's FICO at loan origination belongs to.
fico_range_low	The lower boundary range the borrower's FICO at loan origination belongs to.
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
grade	LC assigned loan grade
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
id	A unique LC assigned ID for the loan listing.
il_util	Ratio of total current balance to high credit/credit limit on all install acct
initial_list_status	The initial listing status of the loan. Possible values are – W, F
inq_fi	Number of personal finance inquiries
inq_last_12m	Number of credit inquiries in past 12 months
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
int_rate	Interest Rate on the loan
issue_d	The month which the loan was funded
last_credit_pull_d	The most recent month LC pulled credit for this loan
last_fico_range_high	The upper boundary range the borrower's last FICO pulled belongs to.
last_fico_range_low	The lower boundary range the borrower's last FICO pulled belongs to.
last_pymnt_amnt	Last total payment amount received
last_pymnt_d	Last month payment was received
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
loan_status	Current status of the loan
max_bal_bc	Maximum current balance owed on all revolving accounts
member_id	A unique LC assigned Id for the borrower member.

Name	**Description**
mo_sin_old_il_acct	Months since oldest bank installment account opened
mo_sin_old_rev_tl_op	Months since oldest revolving account opened
mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened
mo_sin_rcnt_tl	Months since most recent account opened
mort_acc	Number of mortgage accounts.
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_major_derog	Months since most recent 90-day or worse rating
mths_since_last_record	The number of months since the last public record.
mths_since_rcnt_il	Months since most recent installment accounts opened
mths_since_recent_bc	Months since most recent bankcard account opened.
mths_since_recent_bc_dlq	Months since most recent bankcard delinquency
mths_since_recent_inq	Months since most recent inquiry.
mths_since_recent_revol_delinq	Months since most recent revolving delinquency.
next_pymnt_d	Next scheduled payment date
num_accts_ever_120_pd	Number of accounts ever 120 or more days past due
num_actv_bc_tl	Number of currently active bankcard accounts
num_actv_rev_tl	Number of currently active revolving trades
num_bc_sats	Number of satisfactory bankcard accounts
num_bc_tl	Number of bankcard accounts
num_il_tl	Number of installment accounts
num_op_rev_tl	Number of open revolving accounts
num_rev_accts	Number of revolving accounts
num_rev_tl_bal_gt_0	Number of revolving trades with balance >0
num_sats	Number of satisfactory accounts
num_tl_120dpd_2m	Number of accounts currently 120 days past due (updated in past 2 months)
num_tl_30dpd	Number of accounts currently 30 days past due (updated in past 2 months)
num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in last 24 months
num_tl_op_past_12m	Number of accounts opened in past 12 months
open_acc	The number of open credit lines in the borrower's credit file.
open_acc_6m	Number of open trades in last 6 months
open_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
open_il_6m	Number of currently active installment trades
open_rv_12m	Number of revolving trades opened in past 12 months

Name	**Description**
open_rv_24m	Number of revolving trades opened in past 24 months
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
pct_tl_nvr_dlq	Percent of trades never delinquent
percent_bc_gt_75	Percentage of all bankcard accounts >75% of limit.
policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
pub_rec	Number of derogatory public records
pub_rec_bankruptcies	Number of public record bankruptcies
purpose	A category provided by the borrower for the loan request.
pymnt_plan	Indicates if a payment plan has been put in place for the loan
recoveries	post charge off gross recovery
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
sub_grade	LC assigned loan subgrade
tax_liens	Number of tax liens
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
title	The loan title provided by the borrower
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
tot_hi_cred_lim	Total high credit/credit limit
total_acc	The total number of credit lines currently in the borrower's credit file
total_bal_ex_mort	Total credit balance excluding mortgage
total_bal_il	Total current balance of all installment accounts
total_bc_limit	Total bankcard high credit/credit limit
total_cu_tl	Number of finance trades
total_il_high_credit_limit	Total installment high credit/credit limit
total_pymnt	Payments received to date for total amount funded
total_pymnt_inv	Payments received to date for portion of total amount funded by investors
total_rec_int	Interest received to date
total_rec_late_fee	Late fees received to date
total_rec_prncp	Principal received to date
total_rev_hi_lim	Total revolving high credit/credit limit
url	URL for the LC page with listing data.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified

Name	**Description**
verified_status_joint	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.

Appendix B

Univariate plots of variables

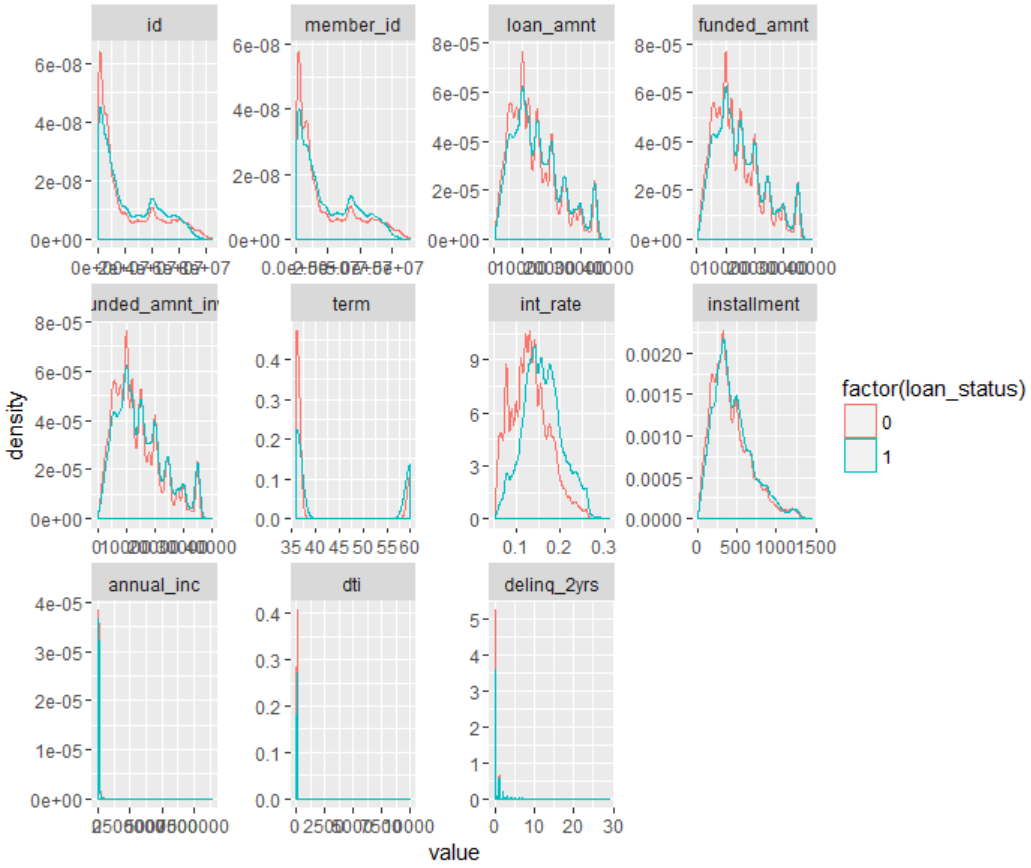


FIGURE B.1: Density plots of numeric features over Loan statuses 1

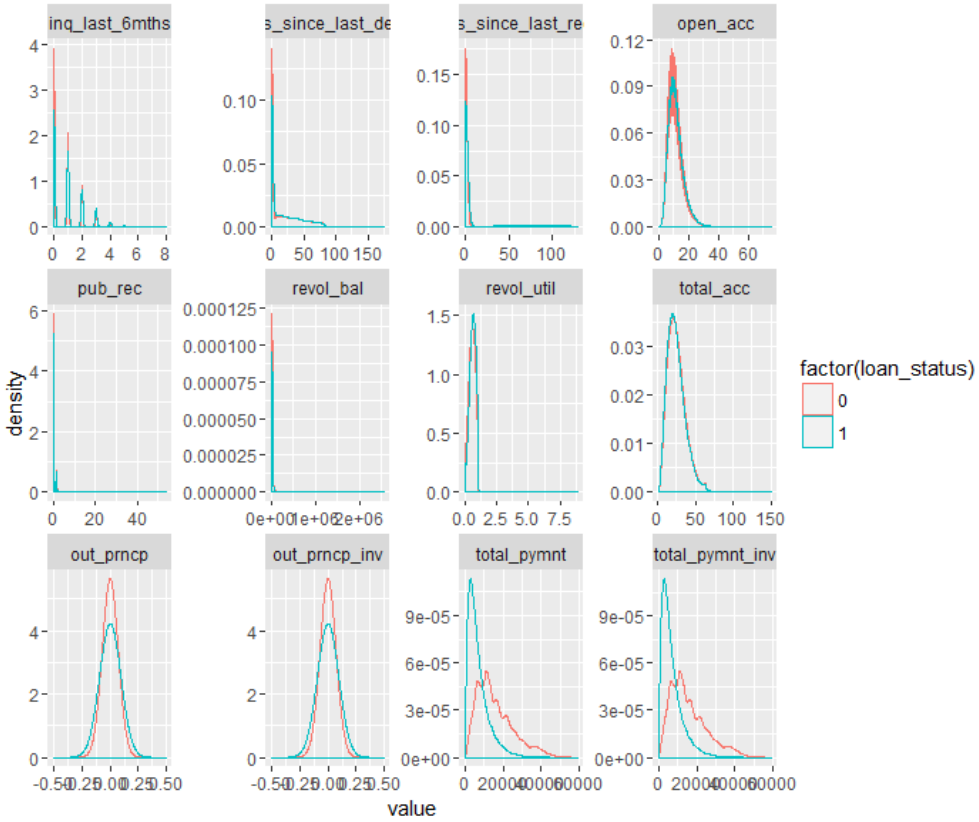


FIGURE B.2: Density plots of numeric features over Loan statuses 2

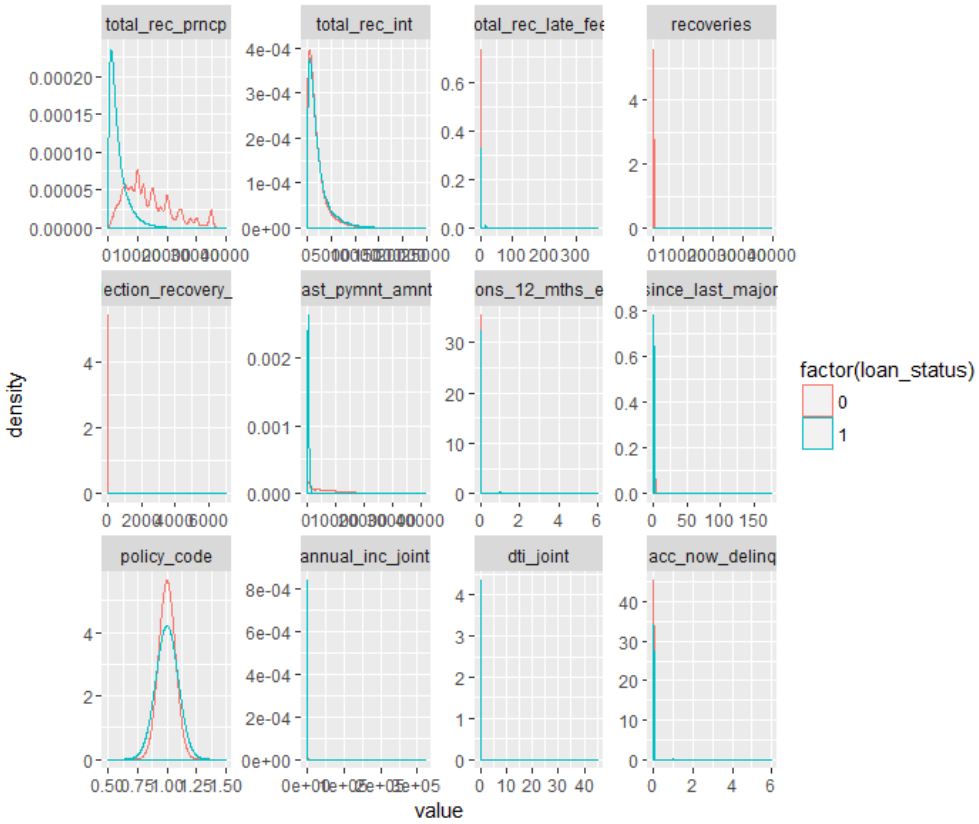


FIGURE B.3: Density plots of numeric features over Loan statuses 3

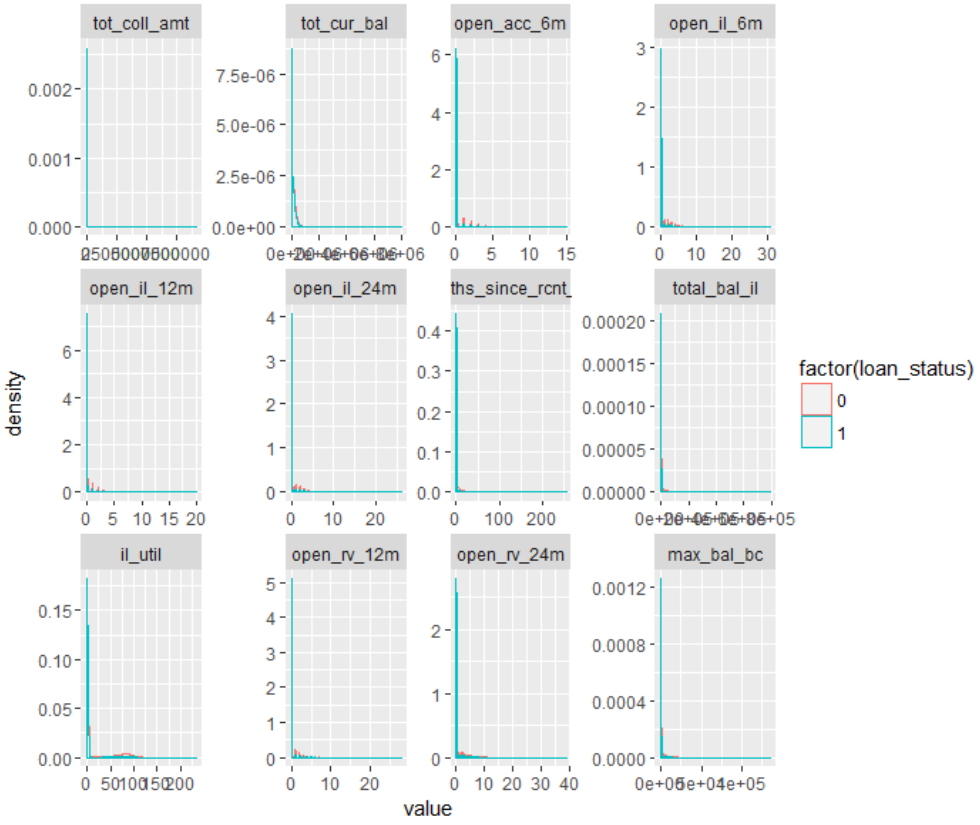


FIGURE B.4: Density plots of numeric features over Loan statuses 4

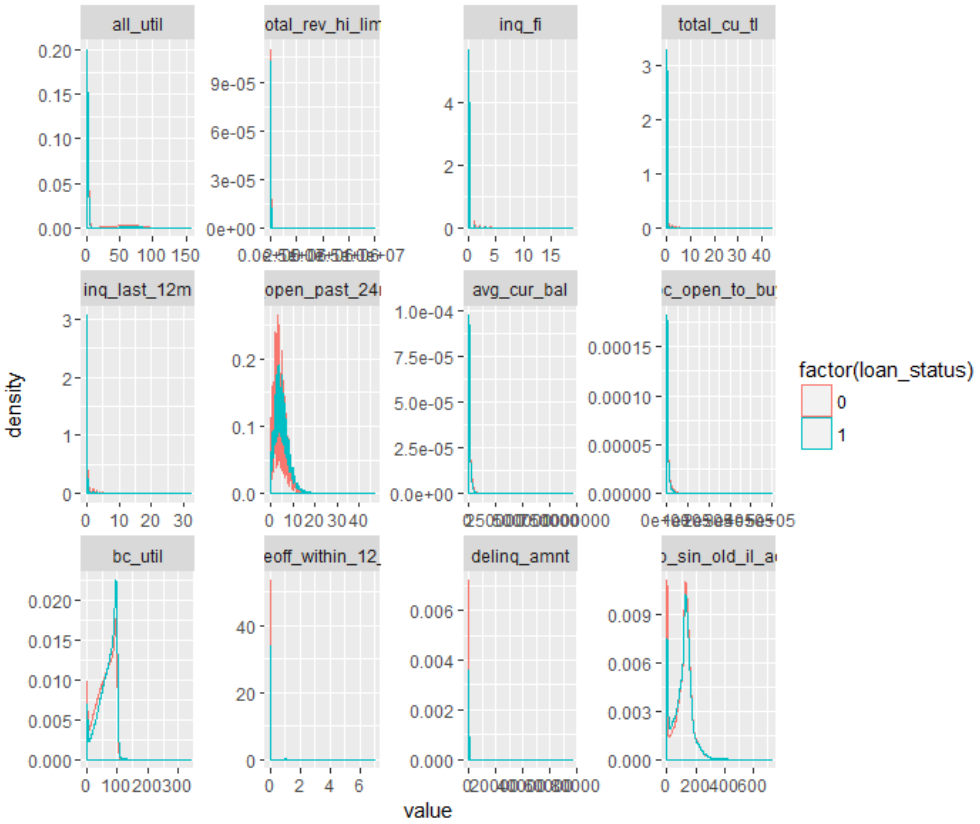


FIGURE B.5: Density plots of numeric features over Loan statuses 5

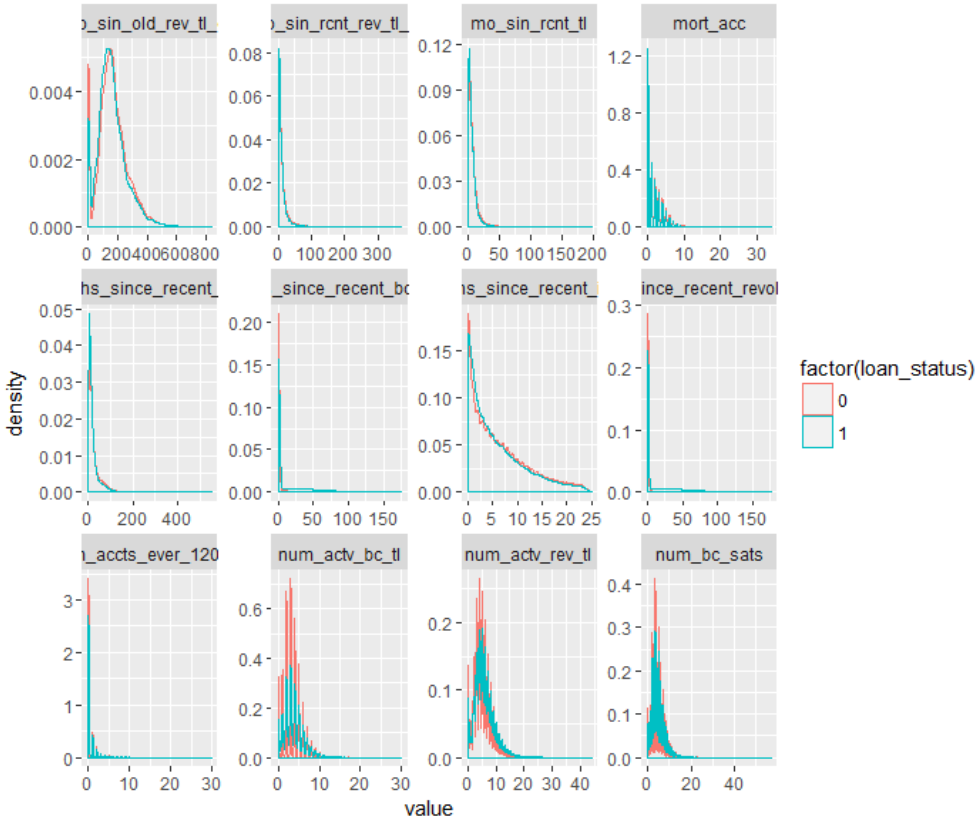


FIGURE B.6: Density plots of numeric features over Loan statuses 6

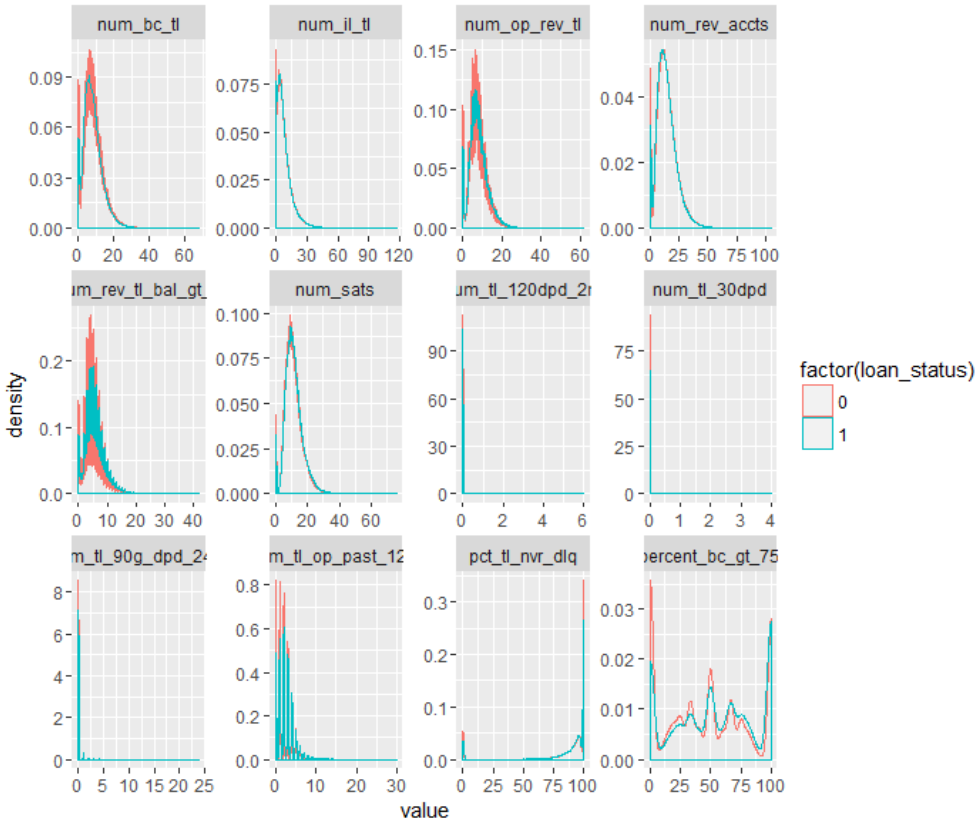


FIGURE B.7: Density plots of numeric features over Loan statuses 7

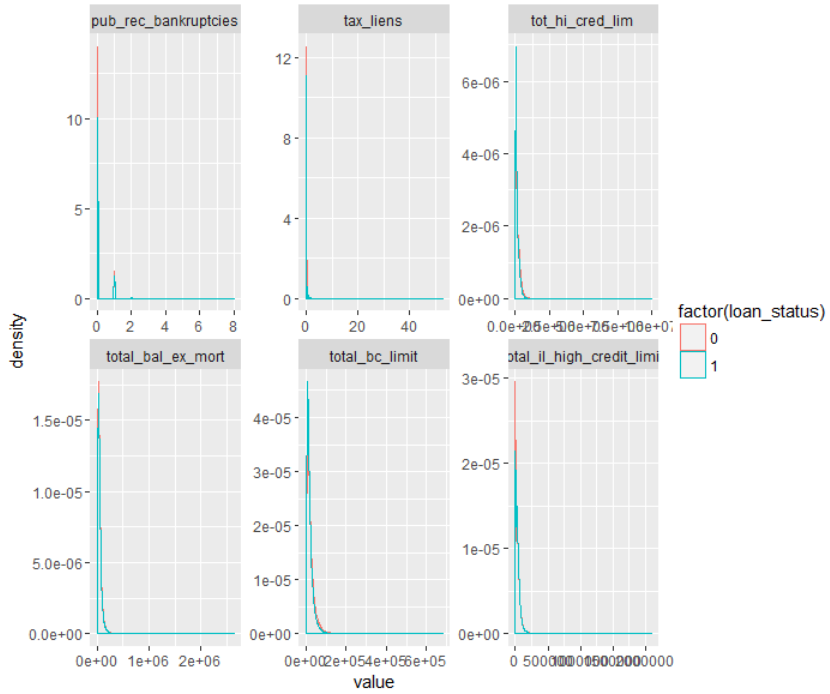


FIGURE B.8: Density plots of numeric features over Loan statuses 8

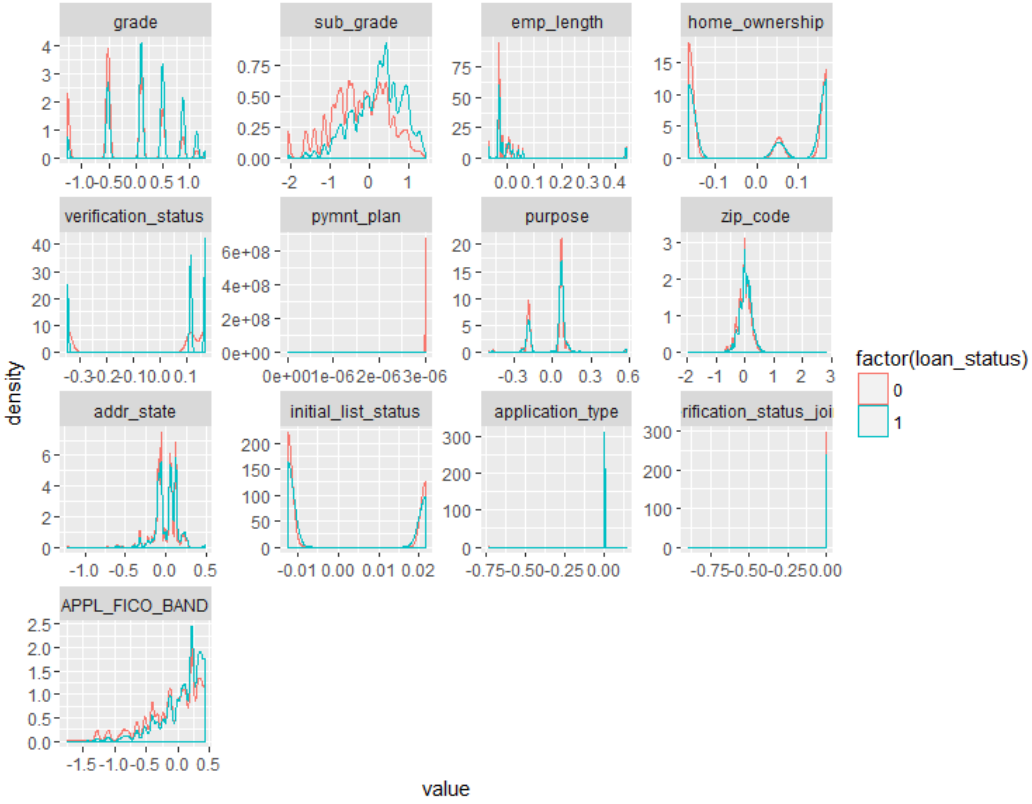


FIGURE B.9: Density plots WOE transformed non numeric features

Appendix C

Descriptive statistics

Variable name	n	mean	sd	min	max	range
loan_amnt	411471	13955.73	8275.54	500.00	40000.00	39500.00
funded_amnt	411471	13929.23	8261.56	500.00	40000.00	39500.00
funded_amnt_inv	411471	13868.81	8273.50	0.00	40000.00	40000.00
term	411471	41.60	10.15	36.00	60.00	24.00
annual_inc	411471	73623.27	62539.79	0.00	8900060.00	8900060.00
dti	411471	17.19	17.70	0.00	9999.00	9999.00
delinq_2yrs	411471	0.27	0.79	0.00	29.00	29.00
inq_last_6mths	411471	0.80	1.04	0.00	8.00	8.00
mths_since_last_delinq	411471	16.06	22.85	0.00	176.00	176.00
mths_since_last_record	411471	10.39	27.87	0.00	129.00	129.00
open_acc	411471	11.21	5.06	0.00	76.00	76.00
pub_rec	411471	0.17	0.49	0.00	54.00	54.00
revol_bal	411471	15698.30	20495.36	0.00	2568995.00	2568995.00
revol_util	411471	0.54	0.25	0.00	8.92	8.92
total_acc	411471	25.30	11.84	2.00	151.00	149.00
collections_12_mths_ex_med	411471	0.01	0.11	0.00	6.00	6.00
mths_since_last_major_derog	372392	10.53	21.61	0.00	176.00	176.00
annual_inc_joint	104720	235.57	5595.28	0.00	320000.00	320000.00
dti_joint	104720	0.04	0.87	0.00	45.39	45.39
acc_now_delinq	411471	0.00	0.07	0.00	6.00	6.00
tot_coll_amt	372392	191.00	15100.50	0.00	9152545.00	9152545.00
tot_cur_bal	372392	129297.51	152265.07	0.00	8000078.00	8000078.00
open_acc_6m	104720	0.18	0.68	0.00	15.00	15.00
open_il_6m	104720	0.41	1.49	0.00	31.00	31.00
open_il_12m	104720	0.14	0.56	0.00	20.00	20.00
open_il_24m	104720	0.29	1.02	0.00	26.00	26.00
mths_since_rcnt_il	104720	2.52	10.90	0.00	255.00	255.00
total_bal_il	104720	5420.23	20792.86	0.00	796104.00	796104.00
il_util	104720	9.38	25.78	0.00	233.80	233.80
open_rv_12m	104720	0.23	0.85	0.00	28.00	28.00
open_rv_24m	104720	0.48	1.57	0.00	39.00	39.00
max_bal_bc	104720	844.07	3010.62	0.00	135471.00	135471.00
all_util	104720	8.55	22.20	0.00	156.90	156.90
total_rev_hi_lim	372392	28499.73	34865.71	0.00	9999999.00	9999999.00
inq_fi	104720	0.17	0.76	0.00	19.00	19.00
total_cu_tl	104720	0.28	1.42	0.00	44.00	44.00
inq_last_12m	104720	0.39	1.42	0.00	32.00	32.00
acc_open_past_24mths	372392	4.50	3.07	0.00	47.00	47.00
avg_cur_bal	372392	12623.40	16126.44	0.00	958084.00	958084.00
bc_open_to_buy	372392	8659.47	13848.54	0.00	497445.00	497445.00
bc_util	372392	61.22	29.22	0.00	339.60	339.60
chargeoff_within_12_mths	411471	0.01	0.10	0.00	7.00	7.00
delinq_amnt	411471	10.06	603.05	0.00	76735.00	76735.00
mo_sin_old_il_acct	372392	112.40	61.36	0.00	724.00	724.00
mo_sin_old_rev_tl_op	372392	165.65	97.56	0.00	842.00	842.00
mo_sin_rcnt_rev_tl_op	372392	11.73	15.06	0.00	372.00	372.00
mo_sin_rcnt_tl	372392	7.22	8.50	0.00	197.00	197.00
mort_acc	372392	1.78	2.16	0.00	34.00	34.00
mths_since_recent_bc	372392	22.75	28.66	0.00	554.00	554.00
mths_since_recent_bc_dlq	372392	9.42	20.29	0.00	176.00	176.00
mths_since_recent_inq	372392	5.85	5.82	0.00	25.00	25.00
mths_since_recent_revol_delinq	372392	11.98	21.40	0.00	176.00	176.00
num_accts_ever_120_pd	372392	0.42	1.15	0.00	30.00	30.00
num_actv_bc_tl	372392	3.36	2.25	0.00	30.00	30.00
num_actv_rev_tl	372392	5.20	3.30	0.00	44.00	44.00
num_bc_sats	372392	4.48	2.83	0.00	57.00	57.00
num_bc_tl	372392	8.14	5.26	0.00	68.00	68.00
num_il_tl	372392	7.85	7.27	0.00	117.00	117.00
num_op_rev_tl	372392	7.64	4.61	0.00	62.00	62.00
num_rev_accts	372392	14.20	8.66	0.00	105.00	105.00
num_rev_tl_bal_gt_0	372392	5.18	3.26	0.00	42.00	42.00
num_sats	372392	10.97	5.49	0.00	76.00	76.00
num_tl_120dpd_2m	372392	0.00	0.03	0.00	6.00	6.00
num_tl_30dpd	372392	0.00	0.06	0.00	4.00	4.00
num_tl_90g_dpd_24m	372392	0.08	0.45	0.00	24.00	24.00
num_tl_op_past_12m	372392	2.00	1.79	0.00	30.00	30.00
pct_tl_nvr_dlq	372392	88.00	25.62	0.00	100.00	100.00
percent_bc_gt_75	372392	47.83	35.71	0.00	100.00	100.00
pub_rec_bankruptcies	411471	0.12	0.35	0.00	8.00	8.00
tax_liens	411471	0.03	0.28	0.00	53.00	53.00
tot_hi_cred_lim	372392	157113.92	170744.90	0.00	9999999.00	9999999.00
total_bal_ex_mort	372392	45538.67	44009.70	0.00	2644442.00	2644442.00
total_bc_limit	372392	19799.51	19912.65	0.00	684000.00	684000.00
total_il_high_credit_limit	372392	35822.85	39913.37	0.00	2101913.00	2101913.00

TABLE C.1: Descriptive statistics numerical features

Appendix D

Random Forest tables

D.1 Variable importance tables

variable	importance	variable	importance
zip_code	416.0527	pct_tl_nvr_dlq	72.68818
APPL_FICO_BAND	213.8836	mths_since_last_delinq	69.57616
addr_state	198.6785	verification_status	66.31102
term	189.9088	inq_last_6mths	65.08286
dti	182.1208	home_ownership	53.16906
annual_inc	154.1063	mths_since_recent_revol_delinq	48.92448
issue_month	149.6464	mths_since_recent_bc_dlq	35.36306
emp_length	145.0916	mths_since_last_major_derog	33.79451
revol_util	135.0623	provide_emp_name	27.87751
total_acc	129.1624	provide_description	25.82997
revol_bal	126.3247	delinq_2yrs	25.00288
bc_open_to_buy	122.2202	diff_fundinvfund	23.8988
mo_sin_old_rev_tl_op	121.8768	tot_coll_amt	22.63836
mths_since_recent_bc	119.0326	mths_since_last_record	22.03393
open_acc	116.7769	pub_rec	21.49708
acc_open_past_24mths	115.7891	pub_rec_bankruptcies	20.14926
total_bc_limit	115.6248	num_accts_ever_120_pd	20.1041
mo_sin_rcnt_tl	112.8249	num_tl_90g_dpd_24m	15.449
bc_util	112.5346	open_il_6m	13.77864
loan_amnt	110.2717	inq_last_12m	12.17467
total_bal_ex_mort	110.1267	mths_since_rcnt_il	11.77917
funded_amnt_inv	109.5812	open_il_24m	11.57781
mo_sin_old_il_acct	108.9638	tax_liens	11.07577
num_il_tl	107.7144	open_rv_12m	11.046
funded_amnt	107.4092	all_util	10.88538
num_bc_tl	106.4846	open_rv_24m	10.55284
avg_cur_bal	105.1786	open_acc_6m	10.13919
num_rev_accts	103.9505	total_bal_il	10.06603
mths_since_recent_inq	103.9079	max_bal_bc	9.765999
total_rev_hi_lim	103.0678	il_util	9.610966
purpose	102.5834	open_il_12m	8.919297
total_il_high_credit_limit	102.3717	total_cu_tl	8.828741
tot_hi_cred_lim	101.9106	inq_fi	8.789708
mo_sin_rcnt_rev_tl_op	99.6896	collections_12_mths_ex_med	7.701438
tot_cur_bal	98.93337	annual_inc_joint	6.778249
num_op_rev_tl	97.86907	dti_joint	6.705896
num_sats	94.27645	chargeoff_within_12_mths	4.499796
percent_bc_gt_75	94.08768	acc_now_delinq	3.919005
num_bc_sats	88.0053	delinq_amnt	2.878546
num_rev_tl_bal_gt_0	84.83888	num_tl_30dpd	2.633626
num_actv_rev_tl	79.8271	num_tl_120dpd_2m	1.942961
num_actv_bc_tl	79.36425	application_type	0.421416
num_tl_op_past_12m	76.27444	verification_status_joint	0.14047
mort_acc	75.74549		

TABLE D.1: Probability of charge off variable importance

variable	importance	variable	importance
term	199.2662	num_actv_bc_tl	20.97415
zip_code	65.89727	num_tl_op_past_12m	20.88428
addr_state	52.3297	tot_hi_cred_lim	20.74125
APPL_FICO_BAND	50.57595	tot_cur_bal	20.13232
issue_month	47.06942	pct_tl_nvr_dlq	19.64586
dti_joint	41.03402	mths_since_last_delinq	19.3337
dti	38.68175	mths_since_rcnt_il	18.90417
emp_length	37.88869	open_il_12m	18.2383
annual_inc	37.24195	inq_last_6mths	18.13466
revol_util	36.90088	num_rev_tl_bal_gt_0	17.91419
funded_amnt_inv	35.97178	num_actv_rev_tl	17.39459
revol_bal	35.61127	mort_acc	16.23404
total_acc	35.48848	open_acc_6m	15.129
annual_inc_joint	34.92662	open_il_6m	14.96836
funded_amnt	34.67011	open_rv_12m	14.5295
loan_amnt	34.62611	home_ownership	13.89173
open_acc	32.96589	mths_since_recent_revol_delinq	12.38545
provide_description	30.7782	open_rv_24m	11.82992
purpose	30.43788	total_bal_il	11.37669
acc_open_past_24mths	29.39274	il_util	10.93223
mths_since_recent_bc	28.64063	all_util	10.42608
total_cu_tl	27.92167	open_il_24m	10.34534
total_bal_ex_mort	27.29522	diff_fundinvfund	9.669767
num_rev_accts	27.0079	inq_last_12m	7.241365
verification_status	26.91651	mths_since_recent_bc_dlq	7.241035
bc_util	26.4554	delinq_2yrs	7.098423
bc_open_to_buy	26.27865	mths_since_last_major_derog	6.878035
total_il_high_credit_limit	26.16477	provide_emp_name	6.463225
mths_since_recent_inq	26.07824	pub_rec	5.329956
total_bc_limit	25.93653	mths_since_last_record	4.848371
max_bal_bc	25.8287	pub_rec_bankruptcies	4.719065
mo_sin_old_rev_tl_op	24.74024	num_accts_ever_120_pd	4.694024
inq_fi	24.2641	tot_coll_amt	4.205073
num_bc_tl	23.51386	num_tl_90g_dpd_24m	2.668991
mo_sin_rcnt_tl	23.50506	tax_liens	1.58807
mo_sin_old_il_acct	23.38268	num_tl_30dpd	0.734865
num_bc_sats	23.15826	chargeoff_within_12_mths	0.660458
num_il_tl	23.12197	collections_12_mths_ex_med	0.657291
num_sats	22.6349	num_tl_120dpd_2m	0.594869
mo_sin_rcnt_rev_tl_op	22.39799	acc_now_delinq	0.308156
num_op_rev_tl	22.29945	delinq_amnt	0.229225
avg_cur_bal	21.81986	application_type	0.042107
total_rev_hi_lim	21.11762	verification_status_joint	0.025409
percent_bc_gt_75	21.07295		

TABLE D.2: Exposure at charge off variable importance

variable	importance	variable	importance
zip_code	18.49764	mort_acc	4.787256
addr_state	15.15434	verification_status	4.645845
APPL_FICO_BAND	14.62438	inq_last_6mths	4.320916
issue_month	11.66216	mths_since_recent_revol_delinq	3.672717
emp_length	10.38577	home_ownership	3.312881
dti	10.23981	dti_joint	2.519129
annual_inc	9.981299	term	2.326144
total_acc	9.893213	annual_inc_joint	2.322649
revol_util	9.749394	mths_since_last_major_derog	2.181642
revol_bal	9.078406	diff_fundinvfund	2.151665
open_acc	8.787862	mths_since_recent_bc_dlq	2.150908
purpose	8.324032	provide_description	2.14102
funded_amnt_inv	8.213143	delinq_2yrs	2.005189
funded_amnt	8.152653	total_cu_tl	1.960205
loan_amnt	8.033118	inq_fi	1.684803
mths_since_recent_bc	7.837823	max_bal_bc	1.609992
mo_sin_old_rev_tl_op	7.766882	open_il_12m	1.35622
bc_open_to_buy	7.586278	num_accts_ever_120_pd	1.35557
acc_open_past_24mths	7.502536	tot_coll_amt	1.29598
num_il_tl	7.479329	provide_emp_name	1.254603
total_bal_ex_mort	7.440958	mths_since_rcnt_il	1.237673
total_bc_limit	7.363676	open_rv_12m	1.153826
mo_sin_old_il_acct	7.359374	pub_rec	1.124605
bc_util	7.147224	mths_since_last_record	1.099333
mo_sin_rcnt_tl	7.073907	open_acc_6m	1.091891
mo_sin_rcnt_rev_tl_op	7.028258	open_il_6m	1.062046
mths_since_recent_inq	7.016301	pub_rec_bankruptcies	1.01377
total_il_high_credit_limit	6.927685	total_bal_il	0.994093
num_bc_tl	6.871056	open_rv_24m	0.948439
num_rev_accts	6.702889	il_util	0.853376
avg_cur_bal	6.662601	num_tl_90g_dpd_24m	0.810522
total_rev_hi_lim	6.605021	open_il_24m	0.76465
num_sats	6.268026	all_util	0.705427
tot_cur_bal	6.166549	inq_last_12m	0.558222
num_op_rev_tl	6.160504	tax_liens	0.557964
tot_hi_cred_lim	6.08958	collections_12_mths_ex_med	0.440338
num_bc_sats	6.014124	chargeoff_within_12_mths	0.18467
percent_bc_gt_75	5.416034	num_tl_30dpd	0.175659
mths_since_last_delinq	5.291957	acc_now_delinq	0.136739
num_actv_bc_tl	5.242819	delinq_amnt	0.089371
pct_tl_nvr_dlq	5.066502	num_tl_120dpd_2m	0.058406
num_rev_tl_bal_gt_0	4.851899	verification_status_joint	0.002298
num_tl_op_past_12m	4.817282	application_type	0.002084
num_actv_rev_tl	4.813762		

TABLE D.3: Loss given charge off variable importance

variable	importance	variable	importance
zip_code	916.0333	mort_acc	242.5059
term	812.2799	verification_status	222.2125
APPL_FICO_BAND	683.3494	mths_since_last_delinq	222.0033
addr_state	637.8747	inq_last_6mths	211.1864
dti	570.3449	home_ownership	178.5671
issue_month	501.993	mths_since_recent_revol_delinq	163.0127
annual_inc	484.9612	mths_since_last_major_derog	107.6556
emp_length	465.9498	mths_since_recent_bc_dlq	105.1092
revol_util	433.4505	diff_fundinvfund	86.32809
total_acc	426.7132	delinq_2yrs	84.75175
revol_bal	397.8141	provide_description	73.80763
mo_sin_old_rev_tl_op	397.35	provide_emp_name	71.788
mths_since_recent_bc	392.926	num_accts_ever_120_pd	71.04579
bc_open_to_buy	386.3735	tot_coll_amt	70.01921
total_bc_limit	374.3714	pub_rec	69.24323
acc_open_past_24mths	374.3121	mths_since_last_record	63.17175
mo_sin_old_il_acct	372.8039	pub_rec_bankruptcies	60.86615
bc_util	364.8361	open_il_6m	55.91736
mo_sin_rcnt_tl	361.8429	num_tl_90g_dpd_24m	49.87108
open_acc	360.2241	mths_since_rcnt_il	41.74651
num_il_tl	355.1622	open_il_24m	40.89717
purpose	354.4437	tax_liens	37.96746
mths_since_recent_inq	351.8612	max_bal_bc	37.79164
mo_sin_rcnt_rev_tl_op	347.8252	inq_last_12m	35.89624
num_bc_tl	347.637	open_rv_12m	33.14875
num_rev_accts	343.7577	open_rv_24m	30.58348
total_bal_ex_mort	341.7942	all_util	30.24407
funded_amnt_inv	337.901	total_bal_il	26.8169
total_il_high_credit_limit	335.0306	total_cu_tl	25.61331
total_rev_hi_lim	331.8602	collections_12_mths_ex_med	24.89331
avg_cur_bal	331.1542	il_util	24.66721
funded_amnt	325.9214	inq_fi	22.39208
loan_amnt	324.2537	open_il_12m	22.30479
num_op_rev_tl	307.5146	open_acc_6m	21.6947
tot_hi_cred_lim	306.494	chargeoff_within_12_mths	16.30801
num_sats	305.7567	annual_inc_joint	13.14243
tot_cur_bal	293.5892	dti_joint	11.83248
num_bc_sats	293.2566	acc_now_delinq	10.77922
percent_bc_gt_75	291.7332	num_tl_30dpd	8.846227
num_actv_bc_tl	262.9294	delinq_amnt	8.68876
num_rev_tl_bal_gt_0	260.1725	application_type	3.622313
num_actv_rev_tl	258.197	num_tl_120dpd_2m	2.848019
num_tl_op_past_12m	249.8079	verification_status_joint	1.083168
pct_tl_nvr_dlq	245.1415		

TABLE D.4: Expected Loss variable importance

D.2 Parameter search tables

mTry	Train MSE	Test MSE	nTree	Train MSE	Test MSE	Node Size	Train MSE	Test MSE
20	0.004904	0.026622	1200	0.006075	0.026822	25	0.015001	0.026787
28	0.004725	0.026636	1400	0.006095	0.026841	40	0.018066	0.02679
30	0.004697	0.026642	1000	0.006077	0.026848	44	0.018649	0.0268
22	0.004846	0.026648	1100	0.006094	0.026849	22	0.014108	0.026806
24	0.004798	0.026653	1500	0.006073	0.026851	27	0.015521	0.026809
29	0.004714	0.026654	900	0.006085	0.026851	31	0.01644	0.026811
26	0.004759	0.026656	1000	0.006082	0.026855	43	0.0185	0.026813
23	0.004821	0.026664	1300	0.006069	0.026865	29	0.015998	0.026815
18	0.004975	0.026669	800	0.006079	0.026865	41	0.018214	0.026816
21	0.004874	0.02667	600	0.006088	0.026877	33	0.016858	0.026818
27	0.004739	0.02667	500	0.006091	0.026882	38	0.017738	0.026818
25	0.004782	0.02667	400	0.006086	0.026882	46	0.018916	0.02682
15	0.00513	0.026701	700	0.006094	0.026891	34	0.017053	0.02682
19	0.004942	0.026704	300	0.006151	0.026957	35	0.017218	0.02682
17	0.005029	0.026706	200	0.006145	0.02696	28	0.015766	0.026822
16	0.005071	0.026721	100	0.006225	0.027043	24	0.01473	0.026825
14	0.00521	0.026745				49	0.019277	0.026827
12	0.005425	0.026757				23	0.014437	0.026829
13	0.005292	0.026781				36	0.017393	0.026829
11	0.005598	0.026819				12	0.010114	0.026832
10	0.005798	0.02684				30	0.016247	0.026833
9	0.006115	0.026842				42	0.018351	0.026834
8	0.006507	0.026931				16	0.011951	0.026834
7	0.007112	0.027056				21	0.013816	0.026835
6	0.007947	0.027193				32	0.016666	0.026835
5	0.00911	0.027388				39	0.017933	0.026837
4	0.01091	0.027638				48	0.019152	0.026838
3	0.014203	0.028092				50	0.019386	0.026838
2	0.021247	0.029222				37	0.017593	0.026839
1	0.029936	0.031233				19	0.013112	0.02684
						47	0.019034	0.026842
						26	0.015272	0.026845
						13	0.010625	0.026846
						45	0.018783	0.026849
						20	0.013484	0.026852
						14	0.011076	0.026854
						18	0.012752	0.026857
						7	0.007291	0.026859
						8	0.007874	0.02686
						15	0.01156	0.026863
						17	0.012376	0.026864
						10	0.009058	0.026864
						6	0.006735	0.026868
						11	0.009585	0.026868
						3	0.004983	0.026873
						5	0.006088	0.026874
						9	0.008493	0.026876
						2	0.004548	0.026885
						4	0.005504	0.026886
						1	0.004291	0.026909

TABLE D.5: Random Forest EAC parameter search

mTry	Train MSE	Test MSE	nTree	Train MSE	Test MSE	Node Size	Train MSE	Test MSE
9	0.002287	0.008203	1400	0.00229	0.008198	49	0.006367	0.008176
8	0.002406	0.008203	1200	0.002284	0.008199	47	0.006301	0.008177
11	0.00211	0.008208	600	0.002301	0.0082	48	0.006333	0.008177
10	0.002192	0.008209	1100	0.002289	0.008201	37	0.005884	0.008179
5	0.003042	0.008209	1300	0.002286	0.008201	44	0.006185	0.008179
7	0.002597	0.008209	900	0.002298	0.008202	42	0.006105	0.00818
14	0.001975	0.00821	1500	0.002289	0.008202	50	0.006401	0.00818
6	0.002802	0.008211	1000	0.002283	0.008206	39	0.005973	0.00818
13	0.001997	0.008212	800	0.002292	0.008206	36	0.00582	0.00818
3	0.004258	0.008215	500	0.002296	0.008207	41	0.006071	0.008181
15	0.001951	0.008215	700	0.002299	0.008208	46	0.006267	0.008183
4	0.003439	0.008215	400	0.002295	0.008216	35	0.005788	0.008183
12	0.00206	0.008217	300	0.002299	0.008228	43	0.006145	0.008183
19	0.001876	0.00822	200	0.002323	0.008245	45	0.006223	0.008184
16	0.001922	0.008221	100	0.002369	0.008276	38	0.005922	0.008184
18	0.001885	0.008223				34	0.005721	0.008185
20	0.001861	0.008224				31	0.005553	0.008185
23	0.001832	0.008225				40	0.006012	0.008186
22	0.001832	0.008225				23	0.004955	0.008186
2	0.006026	0.008226				24	0.005043	0.008187
27	0.001791	0.008226				28	0.005342	0.008187
30	0.001785	0.008227				33	0.005665	0.008188
25	0.001809	0.008231				32	0.00562	0.008188
24	0.001827	0.008232				30	0.005485	0.008189
21	0.001848	0.008232				27	0.005274	0.008189
28	0.001797	0.008233				26	0.005204	0.008189
17	0.001902	0.008235				19	0.004576	0.008189
29	0.001782	0.008237				20	0.004673	0.00819
26	0.001802	0.008237				25	0.005127	0.00819
1	0.008248	0.008268				16	0.004216	0.008192
						22	0.004868	0.008192
						29	0.00542	0.008193
						15	0.004095	0.008194
						21	0.004777	0.008195
						13	0.003796	0.008195
						17	0.004338	0.008196
						14	0.003956	0.008197
						18	0.004459	0.008197
						12	0.003653	0.008198
						3	0.001809	0.008201
						5	0.002309	0.008202
						11	0.003494	0.008202
						8	0.002946	0.008202
						9	0.003162	0.008203
						2	0.001608	0.008206
						10	0.003316	0.008208
						6	0.002515	0.008214
						1	0.001448	0.008214
						4	0.00207	0.008215
						7	0.002739	0.008217

TABLE D.6: Random Forest LGC parameter search

mTry	Train MSE	Test MSE	nTree	Train MSE	Test MSE	Node Size	Train MSE	Test MSE
8	0.017405	0.071639	800	0.016767	0.071558	40	0.047228	0.071421
12	0.015657	0.071672	700	0.016786	0.071606	60	0.052848	0.071429
11	0.015946	0.071673	900	0.016749	0.071618	50	0.050439	0.071432
10	0.016312	0.071674	500	0.016789	0.071632	80	0.056276	0.071457
9	0.01679	0.071681	600	0.016771	0.07165	70	0.054765	0.07147
13	0.01541	0.071683	400	0.016831	0.071712	30	0.042813	0.071472
7	0.018226	0.071705	300	0.016831	0.07172	90	0.057576	0.071481
14	0.015239	0.071718	200	0.016878	0.071881	20	0.036165	0.071483
6	0.01943	0.071727	100	0.017119	0.07214	100	0.058661	0.071497
15	0.015056	0.071764				10	0.025125	0.071579
16	0.014928	0.071769				1	0.010428	0.071679
17	0.014787	0.07179						
18	0.014678	0.07182						
19	0.014579	0.07183						
20	0.014482	0.071863						
21	0.014416	0.071869						
5	0.021052	0.071875						
23	0.014271	0.071876						
22	0.014327	0.071896						
26	0.01408	0.071919						
24	0.014211	0.071923						
25	0.014124	0.071927						
30	0.013887	0.07199						
29	0.013953	0.071995						
27	0.01404	0.071997						
28	0.013984	0.072023						
4	0.023801	0.072085						
3	0.029893	0.072432						
2	0.045883	0.073427						
1	0.071821	0.076389						

TABLE D.7: Random Forest EL parameter search

Appendix E

Regression models

E.1 Exposure at charge off model

```
Call:
lm(formula = model.formula, data = lending_df.train)

Residuals:
    Min      1Q  Median      3Q     Max
-0.94687 -0.07897  0.01446  0.10844  0.48947

Coefficients:
                            Estimate Std. Error t value  Pr(>|t|)
(Intercept)                0.7242840  0.0007728 937.279  < 2e-16 ***
annual_inc_joint           0.8359876  0.0129599  64.506  < 2e-16 ***
term                       0.1577588  0.0017261  91.397  < 2e-16 ***
provide_description        0.1850767  0.0071985  25.710  < 2e-16 ***
inq_fi                    -0.0871134  0.0072945 -11.942  < 2e-16 ***
num_tl_op_past_12m         0.0705302  0.0049749  14.177  < 2e-16 ***
pct_tl_nvr_dlq             0.0726385  0.0125420   5.792 7.01e-09 ***
verification_status        0.0207171  0.0036557   5.667 1.46e-08 ***
dti                        0.0201118  0.0024972   8.054 8.20e-16 ***
mo_sin_old_rev_tl_op       0.0222098  0.0067232   3.303 0.000956 ***
bc_util                    0.0280603  0.0048157   5.827 5.68e-09 ***
APPL_FICO_BAND             0.0253607  0.0027022   9.385  < 2e-16 ***
open_acc                  -0.1914179  0.0154775 -12.368  < 2e-16 ***
diff_fundinvfund           0.0825648  0.0377456   2.187 0.028718 *
total_rev_hi_lim          -0.0253267  0.0065202  -3.884 0.000103 ***
purpose                    0.0507640  0.0046483  10.921  < 2e-16 ***
mort_acc                  -0.0123231  0.0051692  -2.384 0.017131 *
pub_rec                   -0.0330247  0.0208108  -1.587 0.112540
collections_12_mths_ex_med 0.0347500  0.0189359   1.835 0.066490 .
delinq_2yrs                0.0633387  0.0190398   3.327 0.000880 ***
acc_now_delinq             0.1066618  0.0398983   2.673 0.007512 **
inq_last_6mths             0.0579230  0.0052041  11.130  < 2e-16 ***
issue_month               -0.0380450  0.0192951  -1.972 0.048644 *
mths_since_last_delinq    -0.1129974  0.0380246  -2.972 0.002963 **
home_ownership             0.0312624  0.0055828   5.600 2.16e-08 ***
application_type          -0.9537059  0.0585578 -16.287  < 2e-16 ***
revol_util                -0.0253932  0.0052688  -4.820 1.44e-06 ***
revol_bal                 -0.1362097  0.0179903  -7.571 3.75e-14 ***
tax_liens                  0.0283604  0.0191226   1.483 0.138060
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1677 on 53589 degrees of freedom
Multiple R-squared:  0.301,     Adjusted R-squared:  0.3006
F-statistic:   824 on 28 and 53589 DF,  p-value: < 2.2e-16
```

E.2 Loss given charge off model

```
Call:
lm(formula = model.formula, data = lending_df.train)

Residuals:
    Min      1Q  Median      3Q     Max
-1.14511 -0.05507  0.02669  0.07467  0.12208

Coefficients:
                            Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)                0.9214995  0.0004348 2119.129  < 2e-16 ***
annual_inc_joint           0.2786326  0.0071892   38.757  < 2e-16 ***
inq_fi                    -0.0372416  0.0040751   -9.139  < 2e-16 ***
provide_description        0.0181347  0.0039330    4.611 4.02e-06 ***
emp_length                 0.0223883  0.0034351    6.517 7.22e-11 ***
annual_inc                 0.0268224  0.0023547   11.391  < 2e-16 ***
pub_rec                    0.0598971  0.0108261    5.533 3.17e-08 ***
num_actv_rev_tl            0.0116188  0.0031861    3.647 0.000266 ***
```

```
open_acc                  0.0256366  0.0095325    2.689  0.007161 **
home_ownership            0.0090113  0.0027162    3.318  0.000909 ***
term                      0.0033165  0.0010414    3.185  0.001451 **
revol_bal                 0.0504230  0.0101864    4.950  7.44e-07 ***
zip_code                  0.0058918  0.0020570    2.864  0.004182 **
chargeoff_within_12_mths  0.1552672  0.0825920    1.880  0.060123 .
addr_state               -0.0094133  0.0037984   -2.478  0.013206 *
loan_amnt                 0.0077600  0.0033403    2.323  0.020176 *
mths_since_recent_inq    -0.0173259  0.0035959   -4.818  1.45e-06 ***
APPL_FICO_BAND           -0.0070402  0.0015094   -4.664  3.10e-06 ***
total_rev_hi_lim         -0.0100160  0.0035482   -2.823  0.004762 **
mo_sin_old_rev_tl_op     -0.0098088  0.0035167   -2.789  0.005286 **
mths_since_last_delinq    0.0396446  0.0210217    1.886  0.059315 .
application_type         -0.3605040  0.0327670  -11.002  < 2e-16 ***
purpose                  -0.0047549  0.0026027   -1.827  0.067718 .
bc_util                  -0.0053021  0.0026235   -2.021  0.043281 *
delinq_2yrs              -0.0336035  0.0104379   -3.219  0.001286 **
revol_util                0.0055931  0.0029067    1.924  0.054330 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09391 on 53592 degrees of freedom
Multiple R-squared:  0.04925,    Adjusted R-squared:  0.04881
F-statistic:   111 on 25 and 53592 DF,  p-value: < 2.2e-16
```

E.3 Expected loss model

```
Call:
lm(formula = model.formula, data = lending_df.train)

Residuals:
    Min      1Q  Median      3Q     Max
-0.47666 -0.15750 -0.08716  0.00406  1.07571

Coefficients:
                           Estimate Std. Error t value  Pr(>|t|)
(Intercept)                0.1536446  0.0005219 294.392  < 2e-16 ***
term                       0.1407186  0.0014662  95.972  < 2e-16 ***
APPL_FICO_BAND             0.0420352  0.0017083  24.606  < 2e-16 ***
dti                        0.0681909  0.0019808  34.427  < 2e-16 ***
acc_open_past_24mths       0.0959395  0.0032484  29.535  < 2e-16 ***
bc_open_to_buy             0.0374914  0.0027229  13.769  < 2e-16 ***
verification_status        0.0251846  0.0024169  10.420  < 2e-16 ***
avg_cur_bal                0.0299705  0.0029582  10.131  < 2e-16 ***
percent_bc_gt_75           0.0218862  0.0031988   6.842  7.83e-12 ***
zip_code                   0.0839619  0.0021642  38.795  < 2e-16 ***
open_il_6m                 0.0810967  0.0025149  32.246  < 2e-16 ***
annual_inc_joint           0.1148578  0.0073231  15.684  < 2e-16 ***
funded_amnt_inv            0.0940734  0.0042193  22.296  < 2e-16 ***
mort_acc                   0.0199904  0.0037314   5.357  8.45e-08 ***
provide_description        0.0545386  0.0047399  11.506  < 2e-16 ***
revol_util                 0.0134155  0.0035458   3.784  0.000155 ***
annual_inc                 0.0746671  0.0031185  23.943  < 2e-16 ***
mths_since_recent_inq     -0.0398438  0.0054887  -7.259  3.90e-13 ***
total_il_high_credit_limit -0.0818550  0.0064209 -12.748  < 2e-16 ***
purpose                    0.0735547  0.0032650  22.528  < 2e-16 ***
home_ownership             0.0723832  0.0040628  17.816  < 2e-16 ***
inq_last_6mths             0.1196836  0.0043318  27.629  < 2e-16 ***
emp_length                 0.1044092  0.0049609  21.046  < 2e-16 ***
open_acc                   0.0457575  0.0130370   3.510  0.000448 ***
delinq_2yrs                0.1623778  0.0140729  11.538  < 2e-16 ***
pub_rec                   -0.0887399  0.0151692  -5.850  4.92e-09 ***
tax_liens                  0.1120335  0.0147313   7.605  2.85e-14 ***
collections_12_mths_ex_med 0.0696793  0.0152081   4.582  4.61e-06 ***
issue_month                0.0221908  0.0134287   1.652  0.098435 .
total_acc                  0.2693868  0.0117234  22.978  < 2e-16 ***
diff_fundinvfund          -0.0921930  0.0263684  -3.496  0.000472 ***
revol_bal                 -0.1524011  0.0122638 -12.427  < 2e-16 ***
acc_now_delinq             0.0607842  0.0311270   1.953  0.050847 .
mths_since_last_delinq    -0.1473465  0.0279382  -5.274  1.34e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.269 on 279966 degrees of freedom
Multiple R-squared:  0.1115,    Adjusted R-squared:  0.1114
F-statistic:  1065 on 33 and 279966 DF,  p-value: < 2.2e-16
```

Bibliography

BCBS, basel committee on banking (2005). "Working paper 14; Studies on the Validation of Internal Rating Systems".

Breiman, L. (2001a). "Random forests". In: *Machine Learning* 45.1, pp. 5–32.

Breiman, Leo (2001b). "Statistical Modeling: The Two Cultures". In: *Statistical Science* 16.3, pp. 199–215.

Breiman, Leo et al. (1984). *Classification and regression trees*. CRC press.

Brownlee, Jason (2013). *A Tour of Machine Learning Algorithms*. [Online; accessed April 13, 2017]. URL: http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/.

Charpentier, Arthur (2013). *Regression tree using Gini's index*. [Online; accessed April 13, 2017]. URL: https://freakonometrics.hypotheses.org/1279.

Cox, David R (1958). "The regression analysis of binary sequences". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242.

Emekter, R. et al. (2015). "Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending". In: *Applied Economics* 47.1, pp. 54–70.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

Gilardi, Nicolas and Samy Bengio (2000). "Local machine learning models for spatial data analysis". In: *Journal of Geographic Information and Decision Analysis* 4.EPFL-ARTICLE-82651, pp. 11–28.

Gini, Corrado (1912). "Variabilità e mutabilità". In: *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi* 1.

Hu, Shuhua (2007). "Akaike information criterion". In: *Center for Research in Scientific Computation*.

James, Gareth et al. (2013). *An introduction to statistical learning*. Vol. 6. Springer.

Kim, Larsen (2016). *Information: Data Exploration with Information Theory (Weight-of-Evidence and Information Value)*. R package version 0.0.9. URL: https://CRAN.R-project.org/package=Information.

Korotkov, V.B. (2011). *Mercer theorem, Encyclopedia of Mathematics*. URL: http://www.encyclopediaofmath.org/index.php?title=Mercer_theorem&oldid=11889.

LendingClub (2016). *Lending Club Corporation website*. [Online; accessed November 15, 2016]. URL: https://www.lendingclub.com/.

Malley, J.D. et al. (2012). "Probability Machines: Consistent probability estimation using nonparametric learning machines". In: *Methods of Information in Medicine* 51.1, pp. 74–81.

Mateescu, A (2015). "Peer-to-Peer Lending". In: *Data & Society [online]*.

Meyer, David et al. (2017). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-8. URL: https://CRAN.R-project.org/package=e1071.

Nelder, J.A. and R.W.M. Wedderburn (1972). "Generalized linear models". In: *Encyclopedia of statistical sciences*.

OpenCV (2017). *Introduction to Support Vector Machines*. [Online; accessed April 13, 2017]. URL: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.

Qi, Min (2009). *Exposure at default of unsecured credit cards*. Office of the Comptroller of the Currency.

Raghava, G.P.S. (2006). *A SVM-based method for rice blast prediction*. [Online; accessed April 13, 2017]. URL: http://www.imtech.res.in/raghava/rbpred/svm.jpg.

riskarticles.com (2017). *Credit Risk: How to Calculate Expected Loss & Unexpected Loss*. [Online; accessed April 13, 2017]. URL: http://riskarticles.com/credit-risk-how-to-calculate-expected-loss-unexpected-loss/.

Tsai, Kevin, Sivagami Ramiah, and Sudhanshu Singh (2014). "Peer Lending Risk Predictor". In: *CS229 Autumn 2014*.

Wright, Marvin N and Andreas Ziegler (2015). "ranger: A fast implementation of random forests for high dimensional data in C++ and R". In: *arXiv preprint arXiv:1508.04409*.