Credit Risk Evaluation using Machine Learning

Martina Sandberg

Division of Statistics
Department of Computer and Information Science
Linköping University
Supervisors
Per Sidén & Bertil Wegmann

Examiner
Anders Nordgaard
Upphovsrätt


Copyright

The publishers will keep this document online on the Internet – or its possible replacement – for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Martina Sandberg
Abstract

In this thesis, we examine the machine learning models logistic regression, multilayer perceptron and random forests in the purpose of discriminate between good and bad credit applicants. In addition to these models we address the problem of imbalanced data with the Synthetic Minority Over-Sampling Technique (SMOTE). The data available have 273 286 entries and contains information about the invoice of the applicant and the credit decision process as well as information about the applicant. The data was collected during the period 2015-2017. With AUC-values at about 73% some patterns are found that can discriminate between customers that are likely to pay their invoice and customers that are not. However, the more advanced models only performed slightly better than the logistic regression.
Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors at Linköping University Per Sidén & Bertil Wegmann for the continuous support and advise during the research and writing of this thesis.

I would also like to thank PayEx and Dan Rådström for giving me the opportunity to work with them and providing the data for this thesis. Specially I would like to thank my supervisor Mikael Widström at PayEx for providing good support during this work.

Finally, I would like to thank my opponent Kevin Neville for reviewing my thesis and giving valuable improvement suggestions, Caroline Svahn for always being there and making these two years endurable, my family for giving me a place to stay and constant support during this work and my beloved Niklas Holm for all the help and love he gives me.
## Contents

Abstract iv

Acknowledgements vi

1 Introduction 1
   1.1 PayEx .............................. 1
   1.2 Background ........................... 1
   1.3 Previous work ........................ 2
   1.4 Objective ............................ 4

2 Data 6
   2.1 Data description .......................... 6
   2.2 Variable Selection ........................ 6

3 Methods 9
   3.1 Linear Regression .......................... 9
   3.2 Logistic Regression ........................ 9
   3.3 Ordinal Logistic Regression .................... 10
   3.4 Multilayer Perceptron ........................ 10
   3.5 Random Forest ............................ 13
   3.6 Imbalanced Data ........................... 14
      3.6.1 SMOTE ............................... 14
   3.7 Evaluation metrics .......................... 15
      3.7.1 Confusion Matrix ........................ 15
      3.7.2 ROC and AUC ............................ 16
      3.7.3 Variable Importance ....................... 17

4 Results 19
   4.1 Preprocessing of variables ....................... 19
   4.2 Partitions .............................. 19
   4.3 Logistic regression .......................... 21
   4.4 Multilayer perceptron ........................ 23
   4.5 Random Forest ............................ 25
   4.6 Comparison .............................. 27

5 Discussion 29

6 Conclusions and future work 33

Bibliography 34

Appendix 37
   Confusion Matrices ............................ 37
List of Figures

1 Example ROC-curves. .............................................. 16
2 Effect sizes of the estimated parameters in the classification with Logistic Regression for the test data and partition $P_1$. .......................... 21
3 Relative variable importance for the classification with Logistic Regression for the test data and partition $P_1$. ............................. 22
4 Relative variable importance, computed with Olden’s method, for the classification with the multilayer perceptron model for the test data and partition $P_1$. ..................................................... 25
5 Relative variable importance for the classification with Random Forests for the test data and partition $P_1$. ............................. 26
6 Distribution of the variable $CustomerAge$ with distribution of defaults as dark grey. ......................................................... 39
7 Distribution of the variable $MaxLimitReturned$ with distribution of defaults as dark grey. ......................................................... 39
8 Distribution of the variable $Income$ with distribution of defaults as dark grey. .................................................. 40
9 Distribution of the variable $Time$ with distribution of defaults as dark grey. ................................................................. 40
10 Distribution of the variable $InternalDataExist$ with distribution of defaults as dark grey. ................................................... 41
11 Distribution of the variable $PreviousRemindersSent$ with distribution of defaults as dark grey. ................................................... 41
12 Distribution of the variable $City$ with distribution of defaults as dark grey. ................................................................. 42
13 Distribution of the variable $Seller1$ with distribution of defaults as dark grey. ................................................................. 42
14 Relative variable importance, computed with Garson’s algorithm, for the classification with the Multilayer perceptron model for the test data and partition $P_1$. ..................................................... 43
15 ROC-curve for the classifications with Logistic regression and Multilayer Perceptron for the test data and partition $P_1$. .......................... 44
16 ROC-curve for the classifications with Logistic regression and Random forests for the test data and partition $P_1$. .......................... 44
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Variables used in the models.</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Confusion Matrix</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>AUC-values for the classifications with the Logistic regression (LR).</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>AUC-values for the classifications with the Multilayer Perceptron (MLP).</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>AUC-values for the classifications with the Random forest (RF).</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>Sensitivity (TPR) and accuracy (Acc) with the original ($B_0$) test data for classifications with Logistic regression (LR), Multilayer Perceptron (MLP) and Random forests (RF). Values are given in percent.</td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td>Confusion matrix with cost. Values are given in percents.</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>AUC values from Brown and Mues (2012).</td>
<td>31</td>
</tr>
<tr>
<td>9</td>
<td>Sensitivity (TPR) and accuracy (Acc) for classifications with Logistic regression (LR) and Multilayer Perceptron (MLP) from West (2000) and Desai et al. (1996).</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>Confusion matrices for the original ($B_0$) test data with partition $P_1$. Values are given in percents.</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>Minimum value, maximum value, mean and standard deviation for some of the variables.</td>
<td>38</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 PayEx

PayEx is the commissioner of this thesis, they work within the financial sector and offers several payment solutions for channels online and physically (in stores and devices). Several different payment types are supported such as card payments, sms and invoice. PayEx stores information about their customers in their credit systems and information is stored of both individuals and companies.

1.2 Background

Credit scoring is a statistical method used to evaluate the credit risk of loan applications. It aims at predicting the probability that a loan applicant will default (fail to pay their debt) by trying to isolate useful effects of the applicants characteristics on delinquencies and defaults, using historical data. The objective of this credit scoring model is to decide whether the credit applicant is likely to repay the financial obligation and accept the credit or if the credit applicant is likely to default on the financial obligation and deny the credit. The reason for analyzing these kinds of models and attempt to find new ones is so that we can increase the scoring accuracy of the credit decision. According to West (2000), even a fraction of a percent increase in credit scoring accuracy is a significant accomplishment since these small improvements induce future savings.

There are many different techniques used to build a credit scoring model. When credit scoring was first introduced (in the 1950s), statistical discrimination and classification methods were used and according to Thomas et al. (2002) such parametric statistical methods are also commonly used today. The advantages with these methods are that we can find the likely discriminating power of the existing model and the relative importance of the variables used and from this we can remove unimportant variables. As stated in Bishop (2007), in these methods the applicants are classified based on their probability to default on a loan. A threshold is set on the probability to default and the credit is denied if the applicant receives a probability higher than the threshold. Since the assumptions needed for the linear discriminant function are extremely restrictive and do not hold in practice, other regression forms have also been investigated. These have less restrictive rules but still lead to linear scoring rules. The most common one of these is the logistic regression (or logit regression) which is according to Thomas et al. (2002) even today widely used. This model assumes a logistic distribution of the probability to default. Another similar regression model is the probit regression
where the distribution of the probability to default has a cumulative normal distribution. Thomas et al. (2002) also states that non-parametric statistical methods have been used such as classification trees, where the attributes of the applicants are splitted into subgroups which are each classified as satisfactory or unsatisfactory, and also other approaches such as neural networks and support vector machines (SVMs). As described in Bishop (2007), neural networks are non-linear models that classify based on pattern recognition capabilities. From data it tries to learn the relationship between the variables we put into the model and the output target variable. Since no assumptions has to be made regarding the relationship between the inputs and outputs or the distributions of the variables and the errors of the model, this model is more flexible than the previously mentioned models. However, according to several articles on the topic, the neural network tends to have a long training process and it is difficult to identify the relative importance of the input variables and thus it can be hard to understand and interpret the model.

1.3 Previous work

A multitude of research have been done in the field with numerous approaches to the problem of credit scoring. The most frequent used of the different neural network techniques are the multilayer perceptron (MLP) network. This is a feed-forward artificial neural network model which uses the backpropagation algorithm to train the network.

West (2000) compares the MLP model against four other neural network architectures; mixture of experts (MOE), radial basis function (RBF), learning vector quantization (LVQ), and fuzzy adaptive resonance (FAR). The models are tuned using 10-fold cross-validation and are benchmarked against traditional methods as linear discriminant analysis, logistic regression, k-nearest neighbor, kernel density estimation and decision trees. As conclusion the MLP model may not be the optimal model to use while MOE and RBF are good competitors. Also, the logistic regression is found to perform best of the traditional methods.

It is common to come across imbalanced data in credit scoring since the number of defaulters are usually much lower than the number of non-defaulters. Zhao et al. (2015) studies ways to improve the MLP model using three aspects. The first is to optimize the data distribution in data sets and, instead of using the traditional 10-fold cross-validation, a method called Average Random Choosing is applied. The advantage of this method is that it takes the class imbalance in consideration, so the distribution of the instances in each class are balanced in the data sets (training, validation, and test data sets). The second aspect is to compare effects of different training, validation and test instance numbers and the
third aspect is to find the most suitable number of hidden units.

Malhotra and Malhotra (2003) compares the neural networks against multiple discriminant (MDA) models. Results show that no significant improvements are made with respect to identifying good credit loans but for identification of potential default customers the neural network always perform better than the MDA models. The robustness of the model is examined by 7-fold cross-validation.

Abdou et al. (2008) compares neural nets to conventional techniques such as discriminant analysis and logistic regression. Unlike the traditional models the neural nets gave a better average correct classification rate. When to decide which of the neural nets that performs best the preferable decision criterion have to be chosen. If the highest average correct classification rate is preferred the probabilistic neural nets should be used and if the lowest estimated misclassification cost is preferred the best net search MLP with five nodes should be used, according to Abdou et al. (2008).

The problem with model interpretation is considered in Baesens et al. (2003) where three rule extraction techniques are examined; Neurorule, Trepan, and Nefclass. Also ways to visualize the rules are discussed. Conclusions are that both Neurorule and Trepan give good results compared to the logistic regression and could extract very compact rule sets and trees for all data sets.

With neural networks, instead of looking for the single best model West et al. (2005) investigate if ensembles strategies (combine the predictions of a collection of individual models) can provide more accurate generalization. The ensemble strategies considered are: crossvalidation, bagging, and boosting. Results imply that the generalization ability of the neural network ensemble is superior to the single best model.

To find better models for the credit scoring problem support vector machines (SVMs) has also been widely observed. Huang et al. (2004) compare the SVM against the MLP but only small improvements are found. Bellotti and Crook (2009) compare the performance of SVM against linear regression, linear discriminant analysis and k-nearest neighbours. Conclusions state that the SVMs performs equally well as the others and that many support vectors are required to achieve the best performance. When the least square support vector machine (LSSVM) technique is used in Lai et al. (2006) results show that this model outperforms all the benchmark models used. Harris (2015) introduces the use of the clustered support vector Machine (CSVM) and results suggest that this model compare well with nonlinear SVM based techniques in terms of the area under the receiver op-
erating characteristic curve (AUC), while outperforming them in terms of training time. Yu et al. (2010) propose an SVM based ensemble learning approach and results suggest that this model can outperform the other comparable models. Also in the SVM case we have the problem of interpretation. Martens et al. (2007) uses the rule extraction methods Trepan and G-REX and concludes that the SVM rule extraction techniques lose only a small percentage in performance compared to SVMs.

Another approach to the problem of imbalanced data is addressed in Brown and Mues (2012). Here several techniques are compared such as logistic regression, neural networks and decision trees, gradient boosting, LSSVM and random forests. Results indicate that the random forest and gradient boosting classifiers perform very well and are able to cope well with class imbalances in data. With a large class imbalance, the C4.5 regression tree, quadratic discriminant analysis and k-nearest neighbours perform significantly worse than the best performing classifiers.

In Ong et al. (2005), genetic programming (GP) is used and benchmarked against models such as neural networks, decision trees and logistic regression. Results reveal that GP can provide better performance than other models. The authors consider GP more suitable compared to the other models for several reasons, for example, GP can select the important variables automatically and compared to neural networks, GP can determine the adequate discriminant function automatically rather than assign the transfer function by decision-makers.

1.4 Objective

Some of the information usually used to make decisions regarding credits are age, address, income, credit history and geographic statistics. Some internal information at PayEx that can be used to get a better prediction are payment behavior: Has the person payed, when did the person pay, which amount have been payed, was a payment reminder needed, have a debt collection demand been sent or buying behavior: when did the buying occur, at what time and where.

In this thesis we want to use the above internal information and find out which of the variables are useful and to what extent in the purpose of discriminating between good and bad credit applicants. That is, we want to predict from these variables if a person is likely to pay or not and thus we are faced with a binary classification problem.

For this classification problem we want to find a good model and we will com-
pare the performance of the models logistic regression, ordinal logistic regression, multilayer perceptron and random forests. To address the problem of imbalanced data Synthetic Minority Over-Sampling Technique (SMOTE) will be used. The final model should minimize credit risk and at the same time maximize granted credits.
2 Data

2.1 Data description

The data provided by PayEx comes from two sources: PxR and CCP. The PxR data contain information about the invoice of the applicant such as the state of it, distribution and expiring date, amount to pay, amount payed, seller information etc. The CCP data include information about the credit decision process for the applicant and contain information such as date of decision, amount requested by the buyer, amount of credit given, reason for rejection (if rejected) etc. The CCP data set also include information about the applicant such as address, number and age of accounts held, income, payment remarks, gender, amount of active claims and verification checks of the information given by the applicant. In this analysis we will focus on invoice data and only on reservations which are in the state capture, this gives us 273 286 entries in total when the data is merged. The invoices considered are made between the beginning of june 2015 to the end of january 2017.

2.2 Variable Selection

First of all, some exploratory analysis is done on the possible variables. The variables used in this analysis are shown in Table 1. The response variable is defined as “0” if the customer has not defaulted on the invoice, that is, if the invoice is in the claims processes “Invoice” or “Reminder”, and otherwise as “1” (default). So if the invoice is payed after a reminder is sent but before the due date of it the person is labeled as a good customer. This gives us approximately 3.5% defaulted and 96.5% payed invoices. Also, the proportion of reminders sent are approximately 11%.

Each invoice is an observation and there are some customers that have several invoices in this data. Approximately 35% of all customers have two or more invoices and this can be a problem since observations from the same customer can be correlated. One possible solution to this would be to remove some of these observations so that we have maximum of, for example three, invoices per person. Another approach would be to add customer-specific random effects. In this thesis we will instead construct two new variables to take this into account, one which states how many previous invoices the customer has payed in time and the other one states how many invoices the customer has defaulted on before. Conditioned on these, all observations will be seen as independent. Among the 142 sellers this data contains, one is responsible for 73% of all invoices.
The *Income* variable contains some missing values (approximately 26%). This is since when a credit with a small amount is requested, PayEx do not check all information about the customer. To solve this we make this variable categorical, with five states: Missing, 0 - 10000, 10001 - 20000, 20001 - 30000, 30001 - 50000 and > 50000. Other possible solutions is to set all missing values to zero or the mean of all incomes, but in the end we chose to make a linear regression model and with this model predict the *income* of the customer. This is done with customer age and gender as explanatory variables. In addition we create a binary variable which states if the *income* was missing or not. We could also remove all observations where the *income* is missing but applying this does not give better results.

*PreviousRemindersSent* contains the number of reminders sent on previous invoices to the customer. Also here we have missing values, approximately 36%. This can be solved in the same way as mentioned earlier with the *Income* variable. We use the categorical approach with the four states: Missing, 0, 1 - 3 and 4+.

*Time* can be used both as a continuous variable or a categorical with, for example, the states we have chosen to use: 06 - 10, 10 - 14, 14 - 18, 18 - 22, 22 - 02 and 02 - 06. Here many invoices are issued at 03 - 04 and this is since some of the invoices are automatically made because of some subscription. The variable *InternalDataExist* is a logical variable which states whether the person have had an invoice at PayEx before. Other variables used are *Weekday*; which day of the week do the purchase occur, *GrandTotalOutstandingDebt*; debt amount if the person have any debt (collected from several different actors), *MaxLimitReturned*; the maximal credit the person is allowed to take, *AuthorizedAmount*; amount to pay on the invoice, and *City*; if the customer lives in a big city (Stockholm, Göteborg or Malmö) or not.

We will also consider the ratio between the amount on the invoice and the maximal credit allowed and the ratio between income and the maximal credit allowed. If the maximal credit is zero the ratio between payment amount on the invoice and the maximal credit is set to four, which is the maximum value this variable can take and the ratio between monthly income and the maximal credit is set to 1200, which is the maximum value this variable can take.

In Appendix Table 11 the minimum value, maximum value, mean and standard deviation are shown for some of these variables. In Appendix Fig 6-13 the distribution for some of the variables are shown together with distribution of defaults.
Table 1: Variables used in the models.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Type</th>
<th>LR/RF</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>Has the customer payed? (0/1)</td>
<td>Binary</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>AuthorizeAmount</td>
<td>Total payment amount</td>
<td>Float</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>MaxLimitReturned</td>
<td>Max credit</td>
<td>Integer</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>GrandTotalOutstandingDebt</td>
<td>Total outstanding debt</td>
<td>Float</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>RatioCreditPayment</td>
<td>Total payment amount / Max credit</td>
<td>Float</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>RatioCreditIncome</td>
<td>Income / Max credit</td>
<td>Float</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Weekday</td>
<td>Day the purchase is made</td>
<td>String</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Weekday2</td>
<td>Day the purchase is made (Weekday/Weekend)</td>
<td>String</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Purchase time</td>
<td>Integer</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Time1</td>
<td>Purchase time</td>
<td>Categorical</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Income</td>
<td>Monthly income</td>
<td>Float</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Income1</td>
<td>Monthly income</td>
<td>Categorical</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>City2</td>
<td>Customer city (Small or Big)</td>
<td>Categorical</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>CustomerAge</td>
<td>Customer age</td>
<td>Integer</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>CustomerAge2</td>
<td>Customer age squared</td>
<td>Integer</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>InternalDataExist</td>
<td>Do internal data exist (Yes/No)</td>
<td>Binary</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>PreviousRemindersSent</td>
<td>Number of reminders sent on previous invoices</td>
<td>Integer</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>PreviousRemindersSent1</td>
<td>Number of reminders sent on previous invoices</td>
<td>Categorical</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Seller1</td>
<td>Do the invoice belong to the seller responsible for 73% of the invoices</td>
<td>Dummy</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>IncomeM</td>
<td>Is the income missing</td>
<td>Dummy</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td>Gender of the customer</td>
<td>Binary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrevDef</td>
<td>Number of previous invoices the customer has defaulted on</td>
<td>Integer</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>PrevPay</td>
<td>Number of previous invoices the customer has payed in time</td>
<td>Integer</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
3 Methods

3.1 Linear Regression

Linear regression can be defined as

\[ y = w^T X + \epsilon, \]  

(1)

where \( y \) is a vector of response variables, \( X \) a matrix of explanatory variables with an intercept constant, \( w \) a vector of parameters and \( \epsilon \) a vector of error terms. For a more comprehensive account of linear regression see e.g. Bishop (2007).

3.2 Logistic Regression

We consider the logistic regression classification with two classes, \( k = 1 \) for default and \( k = 0 \) otherwise. We know that (as stated in for example Bishop (2007)) by using the logistic sigmoid function

\[ \sigma(a) = \frac{1}{1 + exp\{-a\}}, \]  

(2)

the probability that the applicant have defaulted can be written as a logistic sigmoid acting on a linear function of the explanatory variables as

\[ p(Y = 1|x) = \sigma(w^T x) = \frac{1}{1 + exp\{-w^T x\}}. \]  

(3)

Here \( x \) is a vector containing the explanatory variables and an intercept constant and \( w \) is the coefficient vector describing the variables influence on the class probabilities.
To estimate the parameters of the logistic regression model it is common to use the method of maximum likelihood, which we also apply in this thesis. These maximum likelihood estimates can be found using iterative reweighted least squares (IRLS) and a common method used to do this is the Newton–Raphson method which aims at minimizing the error function. This method updates the parameters by

\[ w^{\text{new}} = w^{\text{old}} - H^{-1} \nabla E(w), \]  

where \( H \) is the Hessian matrix, consisting the second derivatives of \( E(w) \) with respect to the components of \( w \). \( \nabla E(w) \) is the gradient (the vector of partial derivatives), with respect to the parameter \( w \), of the logarithm of the likelihood function which for logistic regression becomes the cross-entropy error function.

### 3.3 Ordinal Logistic Regression

In the ordinal logistic regression the response variable contain more than two classes which have a meaningful sequential order. In addition to the two classes previously used, we include if a reminder was needed on the invoice. So the model is trained with three classes with the order Payed < Reminder < Default. In this way we include the information if a reminder was sent on the current invoice, which is not the case in the other models.

### 3.4 Multilayer Perceptron

The Multilayer perceptron model are composed of a number of neurons (or nodes) which are distributed in an input layer, one or more hidden layers, and an output layer. Each neuron processes its inputs (in an non linear manner) and generates one output value which is transmitted to the neurons in the subsequent layer. Following Baesens et al. (2003), the neurons and layers are arranged in a feedforward manner and feedback connections are not allowed.

As given in Bishop (2007), we consider the input variables \( x_1, \ldots, x_D \), which in the first layer are our selected explaining variables and in the other layers the output value of each neuron in the proceeding layer. We construct \( M \) linear combinations
of these, where $M$ is the number of neurons in the next layer, as

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \quad (5)$$

where $j = 1, \ldots, M$ and the superscript $(1)$ indicates which layer the parameters belongs to, in this case the first. The parameters $w_{ji}$ can be seen as weights and $w_{j0}$ as biases. The bias parameters can be absorbed into the set of weight parameters by defining an additional input variable $x_0$ whose value is $x_0 = 1$, so that (5) takes the form

$$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i. \quad (6)$$

The quantities $a_j$ are known as activations. To determine the neurons output value each of these activations are transformed using a differentiable, nonlinear activation function $h(\cdot)$. The activation functions make it possible for the model to solve nonlinear relations between inputs and outputs. This gives us $z_j = h(a_j)$, called hidden units. The function $h(\cdot)$ is generally chosen to be the logistic sigmoid or the tanh (hyperbolic tangent function), according to Bishop (2007). These values are again linearly combined to give output unit activations

$$a_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j, \quad (7)$$

where $k = 1, \ldots, K$ and $K$ is the total number of outputs. This transformation corresponds to the second layer of the network, and again the bias parameters $w_{k0}$ are absorbed into the set of weight parameters. Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs $y_k$. The activation function used for binary classification problems is the logistic sigmoid function, $y_k = \sigma(a_k)$, where $\sigma(a)$ is defined by (2). By combining these stages we get the overall neural network function.
\[ y_k(x, w) = \sigma \left( \sum_{j=0}^{M} w_{kj}^{(2)} h \left( \sum_{i=0}^{D} w_{ji}^{(1)} x_i \right) \right), \]  

(8)

where the set of all weight and bias parameters have been grouped together into a vector \( w \). Thus the neural network model is simply a nonlinear function from a set of input variables \( \{x_i\} \) to a set of output variables \( \{y_k\} \) controlled by a vector \( w \) of adjustable parameters. The network (8) can be said to be a two-layer neural network and is easily generalized, for instance by considering additional layers.

The neural network learns by repeated adjustment of the weights. The difference between the output of the network and the target output can be seen as an error which we want to minimize. This can be done with the backpropagation algorithm, which starts at the output layer and propagates the error backward through the hidden layers and adjust the weights, usually by the use of some form of gradient descent where the weights are updated according to

\[
 w^{\text{new}} = w^{\text{old}} - \eta \nabla E(w^{\text{old}}),
\]

(9)

where \( \eta > 0 \) is the learning rate. This is done repeatedly until the error is satisfactory small. In summary, the Backpropagation algorithm consists of the following steps, as stated in Bishop (2007).

1. Input a vector of training examples to the network and produce a sequence of weights
2. Forward propagate the input vector through the network in order to generate the networks output values
3. Evaluate the error for all the output units by comparing actual output and target output
4. Backpropagate the errors and adjust the weights to minimize overall error
5. Repeat 1-4 until the performance of the network is satisfactory. (Each repeat is called a training epoch)

The learning rate effects the time needed for the network to converge. If it is too small, too many steps are needed and if it is too large, it will possibly prevent
the error to fall below a certain value. In this thesis we will apply the resilient backpropagation (RPROP) proposed in Riedmiller and Braun (1993) where the problem of setting the learning rate is addressed. This method performs a direct adaptation of the weight-updates based on local gradient information.

3.5 Random Forest

Random forest (RF) is, as stated in Breiman (2001), an ensemble classification technique where many classification and regression trees (CARTs) are grown and the results are aggregated.

As explained in Bishop (2007), when training CART the root node contains all data and for each node all variables are searched to find the best split into two children nodes, each containing one subset of the data according to the splitting rule. This is done until all variables are used and then the tree is pruned based on a criterion that balances residual error against a measure of model complexity.

As can be seen in Breiman (2001), when training the random forests, the root node contains a bootstrap sample of data, with the same size as the original data. Each tree has a different bootstrap sample. Instead of searching through all variables in each node, \( T \) of the variables are selected at random, where \( T \) is a fixed integer and should be much smaller than the number of variables. And when the tree has been grown to maximum size it is not pruned. When a new object is classified each tree gives a classification (vote) and the class that gets the majority vote is chosen.

The splitting criterion used in these two methods is the Gini impurity index

\[
GI = \sum_{k=1}^{K} p_k (1 - p_k),
\]

which is a measure of “node impurity”. Here \( p_k \) is the proportion of data points (in a particular leaf node) assigned to class \( k \). As can be seen, this measure is zero for \( p_k = 0 \) and \( p_k = 1 \) and have a maximum at \( p_k = 0.5 \). This is a measure of how often a randomly chosen observation from the data set would be incorrectly classified if it was randomly classified according to the distribution of classes in the subset. Every split result in a decrease in GI. A low GI (i.e. higher decrease in GI) means that a particular predictor variable plays a greater role in partitioning the
data into the defined classes. One of the variable importance metrics proposed by Breiman (2002) uses the sum of all these decreases in the forest due to the given variable, normalized by the number of trees.

According to Liaw and Wiener (2002), random forests perform very well compared to other classifiers, like discriminant analysis, support vector machines and neural networks and, by the strong law of large numbers, it can be shown that RF always converges and thus is robust against overfitting as shown in Breiman (2001).

### 3.6 Imbalanced Data

The minority class instances often, as in our case, contains the information of interest and thus it is important to correctly classify these. Many times, it is worse to mispredict a rare event than a more common one in the sense of consequences. In addition, misclassification of minority class instances are more likely and many classification algorithms use overall accuracy to optimize and this will make the prediction performance of the minority class worse, as stated in Ertekin et al. (2007).

Among all possible solutions to the problems with imbalanced data, two commonly used methods are sampling based and cost function based. Sampling based methods can be divided into three approaches: oversampling, where more observations are added to the minority class, often by replicating existing observations; undersampling where observations from the majority class are removed; and also a mix of the two have been considered. One example of a method which uses the latter approach is SMOTE (Synthetic Minority Over-Sampling Technique), but instead of replicating minority class instances a new minority class data observation are constructed using an algorithm, that borrows information from neighboring data points. The reason for this approach is that replication of instances can lead to overfitting on these replications according to Chawla (2005).

#### 3.6.1 SMOTE

To handle problems with overfitting and at the same time make the decision region bigger for the minority class, Chawla et al. (2002) introduce SMOTE, a technique that generates synthetic instances examples. This oversampling is done by, for each minority class observation, introduce synthetic examples along the line segments by joining the observations nearest neighbors. In the algorithm, we specify
As the number of nearest neighbors and depending on the amount of extra observations required, some or all of these \( n \) neighbors are randomly chosen. So, for example, if we want to oversample 200\%, two of the \( n \) nearest neighbors are chosen and in each direction one sample is generated, as stated in Chawla et al. (2002). The algorithm generates synthetic samples by taking the difference between the feature vector under consideration and its nearest neighbor. This difference is multiplied by a random number between 0 and 1, and is then added to the feature vector under consideration. This way we get a random point between two specific features according to Chawla et al. (2002). With nominal values a modification of Value Distance Metric is used to find the nearest neighbors and then the majority vote for the values among the nearest neighbors is used according to Chawla (2005).

### 3.7 Evaluation metrics

#### 3.7.1 Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted</td>
<td>0</td>
<td>( TN )</td>
</tr>
<tr>
<td>0</td>
<td>( TN )</td>
<td>( FN )</td>
</tr>
<tr>
<td>1</td>
<td>( FP )</td>
<td>( TP )</td>
</tr>
</tbody>
</table>

One common evaluation metric is the confusion matrix. As can be seen in Table 2, the columns correspond to the actual classes and the rows to the predicted classes. In this analysis, we see defaults as positives (1). In Table 2, \( TN \) (True Negatives) is the number of negative examples correctly classified, \( FN \) (False Negatives) is the number of positive examples incorrectly classified as negative, \( FP \) (False Positives) is the number of negative examples incorrectly classified as positive and \( TP \) (True Positives) is the number of positive examples correctly classified.

Several measures can be calculated from a confusion matrix. For example, **Accuracy** is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications. This is a reasonable measure to use if the data is somewhat balanced, which is not the case here. Since there are so few observations that are in fact positive the matrix will
most likely be skewed and \( TP \) will have few observations, but the Accuracy can
despite this be high, indicating a good fit. In the context of imbalanced data, it
is more appropriate to use the ROC-curve as a measure. Other metrics that can
be calculated are true positive rate (TPR), also known as recall or sensitivity,
which states how many correct positive results occur among all positive exam-
pies. \( TPR = \frac{TP}{TP + FN} \). False positive rate (FPR), also known as fall-out,
states how many incorrect positive results occur among all negative examples.
\( FPR = \frac{FP}{FP + TN} \).

### 3.7.2 ROC and AUC

The Receiver Operating Characteristic (ROC) curve is made by plotting FPR on
the x-axis and TPR on the y-axis. This shows the relative trade-offs between
true positive and false positive predictions. Each instance in the confusion ma-
trix represents one point in the ROC space. Since the output from the scoring
models are probabilities, we specify a certain cutoff parameter. If the predicted
probability of default is greater than this value, we set this instance to default,
otherwise we set it to good. Note that each cutoff is a point on the ROC-curve and
thus gives a different confusion matrix. In this thesis, we will look at the cutoff
which gives the point closest to the TPR of 1 and FPR of 0, evaluated on the test
data, since this cutoff weighs both true positive rate and true negative rate equally.

The best case scenario would be to end up in the upper left corner of the ROC
space, coordinate (0,1) corresponding to the dotted line in Fig. 1. This would mean

![Example ROC-curves](image-url)
that we found a model that can produce a perfect prediction with all positive examples correctly classified and no negative examples misclassified as positive. The worst case scenario would be to end up on (or below) the diagonal line, the dashed and dot-dashed lines in Fig. 1, since that is as good as making a random guess (or worse). So, points above the diagonal line represent classification results better than random and thus good classifications, and points below represent classification results worse than random and thus poor classifications. A commonly used metric is the Area Under the ROC-curve (AUC). The AUC can be computed using the trapezoid method, and according to Hanley and McNeil (1982), this value is the same as the Wilcoxon-Mann-Whitney statistic and Cortes and Mohri (2004) shows that it can be computed as

$$\text{AUC} = \frac{1}{mq} \sum_{i=1}^{m} \sum_{j=1}^{q} 1_{x_i > y_j},$$

where $m$ and $q$ represent the number of positive and negative examples respectively, $x_1, \ldots, x_m$ and $y_1, \ldots, y_q$ the output of the classifier on the positive and negative examples respectively and $1_X$ is the indicator function. A positive feature about this measure is that it is independent of the amount of positive examples and thus it is suitable for evaluating the performance of classifiers on unbalanced data sets.

### 3.7.3 Variable Importance

In the logistic regression, the coefficients can be evaluated by the absolute value of the t–statistic for each estimated parameter. The effect sizes of the coefficients can also be used. For the random forest, as mentioned earlier, the Mean Decrease Gini are used.

In the case with the multilayer perceptron model Garson’s algorithm and Oldens’s method can be used to get relative variable importance. In Garson’s algorithm, Garson (1991) propose that the relative importance of a specific variable can be determined by identifying all weights connecting the specific input node to the output node. The corresponding value, which is the absolute magnitude, describes the relationship between the explanatory variable and the response variable. The Olden method, proposed in Olden et al. (2004), also use the weights connections but uses the sum of the product of raw input-hidden and hidden-output connection weights between each input and output neuron. Compared to Garson’s algorithm
this method has the advantage of keeping the magnitude and sign.
4 Results

The computations in this thesis are done with the use of R (www.r-project.org). To train the Logistic Regression model the package \textit{stats} (R Core Team (2016)) is used, for the MLP \textit{neuralnet} (Fritsch and Guenther (2016)) is used and for RF \textit{randomForest} (Liaw and Wiener (2002)) is used. SMOTE, oversampling and undersampling are done with the package \textit{unbalanced} (Pozzolo et al. (2015)). The package \textit{caret} (Kuhn et al. (2016)) is used to compute the variable importance for the LR and RF and \textit{NeuralNetTools} (Beck (2016)) to compute the variable importance for the MLP. \textit{ROCR} (Sing et al. (2005)) is used to get the ROC-curves and AUC-values.

4.1 Preprocessing of variables

Before further analysis all categorical variables are represented as dummy variables and the continuous variables are normalized according to

$$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}},$$

(12)

so the values are in the range $[0,1]$. To evaluate the variable importance in the logistic regression case with effect sizes we will instead standardize the continuous variables according to

$$\frac{x - \text{mean}(x)}{\text{sd}(x)}.$$  

(13)

so the values have zero mean and unit variance. This is since this gives more interpretable values.

4.2 Partitions

To examine the robustness of the models three partitions of train and test sets are used, 75% and 25% in train and test respectively which will be called $P_1$, 90% and
10% in train and test respectively which will be called $P_2$ and also one where we take half of all default instances in the data and an equal amount of good instances in the train sample which will be called $P_3$. This gives us 3.5% in the train set and 96.5% in the test set. In the first two partitions, there will be an equal amount of defaults in the train and test sets as in the entire data, that is 3.5%. In addition to these cross-validation with five folds will be used which will be called $P_{CV}$.

When using SMOTE three parameters must be set. The number of new instances to generate for each default instance, how many nearest neighbors of these instances for which we will use information and the number of non-default instances to use. We will make six new instances per existing default instance and use the two nearest neighbors for information. Several other number of new instances and number of neighbors have been tested without any change in the results. SMOTE will be used on the train sample and in the cases when the train and test samples are partitioned according to $P_1$ and $P_2$. It will also be applied in cross-validation with five folds. With these training samples two balancing partitions will be tested. Equal number of defaults and good instances which will be called $B_1$ and 20% defaults and 80% good which will be called $B_2$. The unbalanced data will be called $B_0$. 
### 4.3 Logistic regression

![Effect sizes – Logistic Regression](image)

**Figure 2:** Effect sizes of the estimated parameters in the classification with Logistic Regression for the test data and partition $P_1$.

In the logistic regression model we use the variables marked with a star in Table 1, this is since, after testing, these gives best results. In the trained model, almost all of the coefficients have a value below one, as can be seen in Fig. 2. `$PrevPay`, `$Seller1`, `$IncomeM`, `$MaxLimitReturned` and `$City2Small` are among the variables that have a negative sign indicating a negative relation to the outcome. This means that the corresponding odds ratios are smaller than one. So, for example, this means that a higher max credit lower the log odds of default and that it is a somewhat bigger probability to default when you live in a big city and when you buy at a seller who is not seller1. However, all coefficients have small values and many of them are not even significant. The variables that have highest relative importance according to this measure are `$CustomerAge`, `$CustomerAge2`, `$PrevPay`, `$InternalDataExist`, `$ReminderSentMissing`, `$ReminderSent1-3` and `$ReminderSent4+`. And the ones that have the lowest relative importance are `$Income`, `$Time`, `$RatioCreditPayment`, `$AuthorizedAmount`, `$RatioCreditIncome`, `$Weekday` and `$City`.

The relative importance of the variables based on the t-statistic in the model are shown in Fig. 3. This measure indicates that `$PrevDef`, `$PrevPay`, `$Debt` and `$PreviousRemindersSent` are most important while `$Income` and `$Time` among
others seems to have a small predictive ability.

Figure 3: Relative variable importance for the classification with Logistic Regression for the test data and partition $P_1$.

From this model we get the AUC-values as can be seen in Table 3 with corresponding ROC-curve in Appendix Fig. 15. As stated before we want this value to be as close to 100% as possible and if it is 50% or lower we might as well take a random guess. No big differences can be seen between different partitions and data balancing. When using the same variables but changing $Income$ and $Time$ to the continuous ones we get slightly worse AUC-values.
Table 3: AUC-values for the classifications with the Logistic regression (LR).

<table>
<thead>
<tr>
<th></th>
<th>(P_1)</th>
<th>(P_2)</th>
<th>(P_3)</th>
<th>(P_{CV})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B_0)</td>
<td>Train</td>
<td>73.3</td>
<td>73.1</td>
<td>72.8</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>72.7</td>
<td>73.3</td>
<td>73.9</td>
</tr>
<tr>
<td>(B_1)</td>
<td>Train</td>
<td>74.4</td>
<td>74.1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>73.3</td>
<td>73.6</td>
<td>-</td>
</tr>
<tr>
<td>(B_2)</td>
<td>Train</td>
<td>74.1</td>
<td>73.9</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>73.1</td>
<td>73.5</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6 shows percentage of correctly classified instances for different cutoffs. Sensitivity, or TPR in the table, is the percentage of correct classifications of defaults of all defaults and accuracy, or Acc in the table, is the total accurate classified instances among all instances. In this case, the cutoff which corresponds to the point closest to the TPR of 1 and FPR of 0 are 3.5%, with corresponding values marked with bold in Table 6. Two confusion matrices with different cutoffs are shown in Table 10a-10b.

In addition to this binary logistic regression classification, an ordinal logistic regression classification has been applied to the data. After training the model it was evaluated on the two classes Default and Payed. However, there are no improvements in performance with this method and the results are similar to the logistic regression results.

4.4 Multilayer perceptron

In the neural network, the start weights are set to random values and the activation functions and output neuron function are set to the logistic sigmoid. The error function used, which we want to minimize, is the cross-entropy. This is since, according to Bishop (2007) and Venables and Ripley (2002), this is the most appropriate error function to use in the case of classification with binary output. Different values of the stopping criteria for the error function have been tested and in the end, we chose to use the stopping criteria 10 for the original data and 100 for the balanced data. This is since a smaller value did not get a better prediction performance, in fact the network started to become overfitted, and a greater value gave worse prediction performance. The training of the neural networks is
done using resilient backpropagation, explained in Riedmiller and Braun (1993). Also, we will have to set the number of hidden layers and the number of hidden neurons, which can have a great impact on the performance of the network. While several different combinations of layers and neurons have been tested, we give results for one hidden layer and 10 neurons here. The network is trained two times to get a stable result.

In the MLP model we use the variables marked with a star in Table 1. Here we use *Time* and *Income* as continuous values since this gives somewhat better results. From this we get the AUC-values as can be seen in Table 4 with corresponding ROC-curve in Appendix Fig. 15. As with the logistic regression, we do not see any major differences in values for different partitions and data balances.

Table 4: AUC-values for the classifications with the Multilayer Perceptron (MLP).

<table>
<thead>
<tr>
<th></th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
<th>$P_{CV}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0$</td>
<td>Train</td>
<td>75.0</td>
<td>74.8</td>
<td>75.0</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>73.9</td>
<td>74.0</td>
<td>73.3</td>
</tr>
<tr>
<td>$B_1$</td>
<td>Train</td>
<td>74.5</td>
<td>74.3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>73.0</td>
<td>73.6</td>
<td>-</td>
</tr>
<tr>
<td>$B_2$</td>
<td>Train</td>
<td>75.7</td>
<td>75.6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>73.9</td>
<td>74.1</td>
<td>-</td>
</tr>
</tbody>
</table>

The percentage of correctly classified instances for different cutoffs are given in Table 6. In this case, the cutoff which corresponds to the point closest to the TPR of 1 and FPR of 0 are 3.3%. Marked with bold in Table 6 are the values closest to this point. Two confusion matrices with different cutoffs are shown in Table 10c-10d.

Relative importance of the variables based on Garson’s algorithm are shown in Appendix Fig. 14 and based on model weights (Olden) is shown in Fig. 4. Both imply that *PrevDef*, *PrevPay* and *Debt* are important variables.
Figure 4: Relative variable importance, computed with Olden’s method, for the classification with the multilayer perceptron model for the test data and partition $P_1$.

### 4.5 Random Forest

In the random forests classification, only two parameters have to be set. The number of trees and the number of variables randomly chosen in each split. A great amount of trees should be used so we use 5000 trees and after some testing we use 10 randomly chosen variables. Bootstrap sampling is a problem in the case of imbalanced data since it would, with a great probability, select only instances from the majority class. To deal with the class imbalance in this model we use stratified sampling without replacement. This means that we divide the classes and pick a selected amount of instances from each. In this thesis, we pick 1000 from each since with testing this gives the best result.

Here we use the same variables as in the logistic regression, since this gives best results. We get the AUC-values as can be seen in Table 5 with corresponding ROC-curve in Appendix Fig. 16 and can conclude that the values are similar to the ones for the previous approaches with the exception for the train values which are somewhat higher.
Table 5: AUC-values for the classifications with the Random forest (RF).

<table>
<thead>
<tr>
<th></th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
<th>$P_{CV}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>83.1</td>
<td>81.8</td>
<td>95.0</td>
<td>82.6</td>
</tr>
<tr>
<td>Test</td>
<td>73.9</td>
<td>74.0</td>
<td>73.0</td>
<td>73.7</td>
</tr>
</tbody>
</table>

Table 6 shows percentage of correctly classified instances for different cutoffs. In this case, the cutoff which corresponds to the point closest to the TPR of 1 and FPR of 0 are 50%, with corresponding values marked with bold in Table 6. Two confusion matrices with different cutoffs are shown in Table 10e-10f.

The relative importance of the variables based on the Mean Decrease Gini are shown in Fig. 5. Mean Decrease Gini gives $\text{RatioCreditIncome}$, $\text{RatioCreditPayment}$ and $\text{AuthorizedAmount}$ as most important variables and the $\text{Income}$ and $\text{Time}$ variables least important among others.

![Variable importance - RF](image)

Figure 5: Relative variable importance for the classification with Random Forests for the test data and partition $P_1$. 

4.6 Comparison

As can be seen in Tables 3, 4 and 5 the AUC-values are very similar both between models but also within the models with different partitions and data balance. It seems that MLP and RF are only slightly better than the LR, which can also be seen in Appendix Fig. 15-16 where the ROC-curves follows almost the exact same path. In these figures, the dots corresponds to the cutoff 3.5% for LR, 3.3% for MLR and 50% for RF. These are a kind of optimal cutoff with respect to both TPR and FPR. The corresponding values for these can also be seen as bold in Table 6. We want the TPR to be as high as possible, but at the same time we do not want to misclassify too many of the good customers. When we lower the cutoff, more defaults are correctly classified but the total accuracy of the model decrease. This can also be observed in the confusion matrices in Appendix Table 10 where, when we set a lower cutoff, more customers that have defaulted are also classified accordingly but it also causes many of the good customers to be wrongly classified as defaults. Also in these matrices, we see that the models predictive performance are similar.

Table 6: Sensitivity (TPR) and accuracy (Acc) with the original (B₀) test data for classifications with Logistic regression (LR), Multilayer Perceptron (MLP) and Random forests (RF). Values are given in percent.

<table>
<thead>
<tr>
<th>Cutoff</th>
<th>2</th>
<th>3</th>
<th>3.5</th>
<th>40</th>
<th>50</th>
<th>55</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>TPR</td>
<td>Acc</td>
<td>TPR</td>
<td>Acc</td>
<td>TPR</td>
</tr>
<tr>
<td>LR-P₁</td>
<td>34.8</td>
<td>89.7</td>
<td>58.3</td>
<td>74.2</td>
<td>66.8</td>
<td>66.2</td>
</tr>
<tr>
<td>LR-P₂</td>
<td>34.0</td>
<td>90.4</td>
<td>58.3</td>
<td>75.2</td>
<td>66.9</td>
<td>66.9</td>
</tr>
<tr>
<td>MLP-P₁</td>
<td>42.2</td>
<td>87.2</td>
<td>63.1</td>
<td>71.1</td>
<td>69.9</td>
<td>63.5</td>
</tr>
<tr>
<td>MLP-P₂</td>
<td>41.5</td>
<td>87.7</td>
<td>63.5</td>
<td>70.1</td>
<td>70.8</td>
<td>63.8</td>
</tr>
<tr>
<td>RF-P₁</td>
<td>3.47</td>
<td>100</td>
<td>3.47</td>
<td>100</td>
<td>3.48</td>
<td>100</td>
</tr>
<tr>
<td>RF-P₂</td>
<td>3.47</td>
<td>100</td>
<td>3.47</td>
<td>100</td>
<td>3.50</td>
<td>100</td>
</tr>
</tbody>
</table>

When choosing the cutoff, the question is how much do we value to correctly classify defaulters above incorrectly classify good customers as defaulters. In credit scoring there is usually a higher cost to give credits to customers who will not pay than to deny credit to a customer who will pay. But at the same time if too many customers are lost it will decrease the revenue for the seller. We can see in Table 6 that if we correctly classify approximately 90% of the defaulters we get an overall accuracy of approximately 40%. This would be a good cutoff if a defaulting customer entails a really high cost, otherwise it is not since the accuracy is very low. As mentioned before, one way to find an “optimal” cutoff is to find the one that gives the point closest to the TPR of 1 and FPR of 0. This is
since this cutoff weigh both true positive rate and true negative rate equally. This
gives about 65% for both TPR and total accuracy for all models used in this thesis.

Table 7: Confusion matrix with cost. Values are given in percents.

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Predicted</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>38.7</td>
</tr>
<tr>
<td>1</td>
<td>57.8</td>
</tr>
</tbody>
</table>

We can for example set the cost of falsely classifying the customer as good, $FN$
in Table 2, to 50 and the cost of falsely classifying the customer as default, $FP$ in
Table 2, to 1. This means that it costs 50 times more to misclassify a defaulting
customer than misclassify a non-default customer. If we apply this to the logistic
regression classification with partition $P_1$ we get that 2.3% is the “optimal” cutoff
with a corresponding accuracy of 41.7%, sensitivity of 86.2% and confusion matrix
as given in Table 7.

From Fig. 3, 4 and 5 we can see that LR and MLP seems to be more similar with $PrevDef$, $PrevPay$, $Debt$ and $PreviousRemindersSent$ as most important while
RF value the two ratio variables and $AuthorizedAmount$ more. The $Income$ and
$Time$ variables are among the lowest valued in all models together with $Weekday$,
$City$, $MaxLimitReturned$ and $IncomeM$. However, some of the time intervals
imply a higher importance than others. We can also see that LR and MLP differ in
the least important variables where $InternalDataExists$, $CustomerAge$, $Seller1$
and $MaxLimitReturned$ have very small weights in MLP but not in LR. The
RF model agrees with a low weight on $InternalDataExist$ and $Seller1$ but have
higher weights on the others. $CustomerAge$2 are quite important in all models.
5 Discussion

Initially, many things can be done differently in the parameter selection part of this thesis. Concerning the income variable, except for the categorical and the regression solution we have tried to set all missing values to zero or the mean of all incomes and also to remove all these observations, but these methods gave worse results. The choice to use linear regression as a predictor of income, with gender and age as explanatory variables, may not be the optimal solution either, and we cannot check if we get the correct income, but it is the best solution of the ones we have tested. However, this is not a further issue for PayEx since this is information they can obtain for all applicants.

The seller that is responsible for 73% of all invoices in the data have a subscription services so invoices are sent out automatically. This is also the reason why so many invoices are issued at 03-04. This can result in a somewhat biased result. Analysis have been done with the data when all instances with invoices from this seller is removed. Somewhat worse results are obtained and no difference in the pattern between the models are shown.

As Zhao et al. (2015) we have tried several different distributions and partitions in the train and test sets but without any obvious differences. Also, instead of normalization, we have tried to standardize the variables so that each have zero mean and unit variance without any bigger change in results.

A lot of research have been done that consider the imbalance problem but usually not with data as imbalanced as the one we use here. Popular data sets used are the German credit scoring data and the Australian credit scoring data which can be found at the UCI Machine Learning Repository. The German data have 1000 observations in total and 30% defaults and the Australian have 690 observations and 55.5% default customers. In this thesis we have 273 286 observations, which is a lot compared to many other articles and in addition we have a very small amount of defaults. It is really unexpected that SMOTE did not give more improvements as it gave very little increase in performance or none at all as can be seen in Tables 3, 4 and 5. We have tried different parameter settings and with all models but without success. Also, random oversampling and undersampling have been applied without better results.

Many other articles find that, among the traditional methods, the logistic regression perform best. Although we do not have too much to compare with, this results also suggests this since we get almost the same results for this simple model as with the more advanced ones.
Since the \textit{PreviousRemindersSent} variable seems to be significant in the binary classification models we expect the ordinal logistic regression to give a better result. It should capture the characteristics of the customers that do not pay right away. Unfortunately, this model did not perform better than any of the others even though we tried different explanatory variables and different balancing in the data.

The random forests model is easy to implement with few parameters to set and, according to Liaw and Wiener (2002), often not so sensitive to these settings. Some different parameter settings have been applied and the ones used are the best performing of these tested, although the differences are not big. Also, the balances \( B_1 \) and \( B_2 \) have been applied but did not perform as good as the stratified sampling.

The MLP model have been trained with several different parameter settings. Different amount of layers and neurons, different learning algorithms, activation functions and variables but without success. It is unexpected that this model gave as small improvements as it did compared to the logistic regression. According to Zhao et al. (2015), if too few hidden neurons are selected the network cannot solve complex problems and if too many are selected it can result in low efficiency and accuracy, however this does not seem to affect our data considerably. It is interesting that the difference between using one and several layers as well as few and many neurons gives about the same results. According to some research, for example Baesens et al. (2003), one hidden layer is often sufficient to give a good model for prediction. It seems that this can be applied to the current data as well. This result also suggests that the connection between the explanatory variables and the response variable does not have a complex relationship so that the LR and the MLP can identify the same patterns.

With neural networks, there exists many different learning algorithms. Improvements of the backpropagation as well as other techniques. Here we have chosen to use the resilient backpropagation, proposed by Riedmiller and Braun (1993), since it is faster than the original backpropagation algorithm and a learning rate is not needed. Many different activation functions can be used and, at least in the output-unit activation function case, the optimal to use according to Bishop (2007) is the logistic sigmoid and thus, this activation function have been used for all neurons. Other activation functions have also been tested without success. The initial weights can be set in many different ways but are usually set to small random values, which we have decided to do as well. The network can be trained several times to be more robust and here we only train it two times since the network training usually takes a long time. It seems that the network is easily
overtrained and a solution to this is to use drop out, which means that in each back-propagation some of the weights are ignored and kept constant. However, no improvements are made when applying this in our network training.

Brown and Mues (2012) present AUC values at varying degrees of class imbalances on different data sets. Note that these data sets are much smaller than the one we use in this thesis. Among the classification models used the LR, MLP and RF are included. In the case with 2.5% defaults, the AUC for the test set for five data sets can be seen in Table 8. Overall we get better AUC-values with our data than any of these five data sets, which can be seen in Tables 3, 4 and 5. It is also obvious that none of these models outperform the other ones in all cases and also that the values can vary more or less between the models.

Table 8: AUC values from Brown and Mues (2012).

<table>
<thead>
<tr>
<th></th>
<th>Data1</th>
<th>Data2</th>
<th>Data3</th>
<th>Data4</th>
<th>Data5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>72.7</td>
<td>73.9</td>
<td>55.1</td>
<td>50.0</td>
<td>50.0</td>
</tr>
<tr>
<td>MLP</td>
<td>71.2</td>
<td>70.2</td>
<td>59.2</td>
<td>70.0</td>
<td>62.3</td>
</tr>
<tr>
<td>RF</td>
<td>69.2</td>
<td>71.3</td>
<td>69.1</td>
<td>87.9</td>
<td>68.7</td>
</tr>
</tbody>
</table>

Also in West (2000) and Desai et al. (1996) the LR and MLP are included. Here they do not use the AUC as measure but percentages of correctly classified instances. In Table 9 the total percentage of instances correctly classified (Acc) and the percentage of defaults correctly classified (TPR) are shown for the two. West (2000) averages over two data sets with 3% and 65% bad loans respectively and Desai et al. (1996) averages over three data sets with 18.42%, 25.98% and 21.15% bad loans respectively. Note that, also here, these data sets are much smaller than the one we use in this thesis. These values can be compared to the values in Table 6 and it is obvious that the former are better since we cannot get an accuracy of around 80% together with a TPR of 45% or 67%. Also, here we can see that the results between the models are similar.

Table 9: Sensitivity (TPR) and accuracy (Acc) for classifications with Logistic regression (LR) and Multilayer Perceptron (MLP) from West (2000) and Desai et al. (1996).

<table>
<thead>
<tr>
<th></th>
<th>West</th>
<th>Desai</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>TPR</td>
</tr>
<tr>
<td>LR</td>
<td>81.78</td>
<td>67.29</td>
</tr>
<tr>
<td>MLP</td>
<td>81.36</td>
<td>67.83</td>
</tr>
</tbody>
</table>
Some effort has been made to estimate a SVM model. These models have been proven to be very time consuming and have not resulted in any improvement in results compared to the models already analyzed.

When evaluating the variable importance plots it is important to note that Garson’s algorithm may give misleading results. This is since it uses the absolute values of the weights and thus it does not provide the direction of the relationship between the input and output variables. So, the weights that change sign between the input and output layers will have a canceling effect which is not included in this algorithm.

From these models, we get somewhat different results regarding the importance of the variables used. However, all seems to agree that Time and Income are among the least important ones together with Weekday, City, MaxLimitReturned and IncomeM. Some of the time intervals imply a higher importance than others. This could be explained by the uneven distribution of this variable with many invoices issued at one time. From these results, it is obvious that the time, place, income and maximum credit the customer can take do not affect if the customer will default or not.

All used models agree that PrevDef, PrevPay and GrandTotalOutstandingDebt are among the most important variables. So, it is obvious that how the customer have behaved on previous invoices and existing debt affect the prediction performance of the models.

In addition to these the MLP model and LR model have PreviousRemindersSent among the most important which the RF model does not. The LR model and the RF model have CustomerAge and CustomerAge2 among the most important which the MLP model does not. The LR model is the only one that values InternalDataExist and Seller1 while the RF model is the only one that values RatioCreditIncome, RatioCreditPayment and AuthorizedAmount.
6 Conclusions and future work

In this thesis logistic regression, ordinal logistic regression, multilayer perceptron and random forests were used to predict if a person will pay their invoice or not. No obvious differences are found between the models and with AUC-values around 73% the models are not very good at discriminating between good and bad customers. SMOTE has also been applied to the train data without better classification results. From these results, it seems that PayEx have already done a good job at eliminating defaulters at the invoice acceptance stage, since this may infer that it is difficult to find patterns that can further discriminate between good and bad customers, in the data we have been given. Since the models perform almost equally well the logistic regression is preferred since it is faster, easier to use and interpret.

However, some patterns are found and it seems that all models agree that the previous behavior of the customer such as the number of previous invoices the customer have defaulted on, the number of previous invoices the customer have payed in time and the number of reminders sent on previous invoices are important predictors in the classification as well as the total outstanding debt the customer has. All models also agree that the time and weekday the buying occur, the customer income and the city the customer lives in are among the least important variables. Apart from these the importance of the variables differ among the models.

Since these results are somewhat disappointing, further research should be done on the neural network field, with other neural network architectures instead of MLP as, for example, West (2000) shows that this may be advantageous. Also in the SVM field where the least square support vector machine technique, used in Lai et al. (2006), and the clustered support vector machine, used in Harris (2015), seem to perform better than other SVM based techniques as well as other models. It would also be interesting to address the problem with model interpretation, as considered in Baesens et al. (2003) and Martens et al. (2007). Obviously, it would only be interesting if a better model could be produced. Further research could also include more ensemble strategies since as shown in West et al. (2005) and Yu et al. (2010), this can give better results. Genetic programming would also be interesting to apply since Ong et al. (2005) concludes that this method can provide better performance than other models. To optimize parameters is also a field that we could do further research, as in Zhou et al. (2009). There are several other algorithms to deal with the problem of unbalanced data that could be tested and also, several other possible variables can be tested to find the ones that can do a better job at explaining the difference between defaulting and good customers.
Bibliography


Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi. un-


Appendix

Confusion Matrices

Table 10: Confusion matrices for the original \((B_0)\) test data with partition \(P_1\). Values are given in percents.

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1</td>
<td></td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>Pred.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>64.5 1.17</td>
<td>0</td>
<td>31.7 0.36</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>32.0 2.30</td>
<td>1</td>
<td>64.8 3.11</td>
<td></td>
</tr>
</tbody>
</table>

(a) LR - Cutoff: 3.5

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1</td>
<td></td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>Pred.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>65.7 1.17</td>
<td>0</td>
<td>39.2 0.44</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30.8 2.29</td>
<td>1</td>
<td>57.3 3.03</td>
<td></td>
</tr>
</tbody>
</table>

(c) MLP - Cutoff: 3.3

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1</td>
<td></td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>Pred.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>66.0 1.17</td>
<td>0</td>
<td>24.8 0.19</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30.5 2.30</td>
<td>1</td>
<td>71.8 3.28</td>
<td></td>
</tr>
</tbody>
</table>

(e) RF - Cutoff: 50

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1</td>
<td></td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>Pred.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>65.7 1.17</td>
<td>0</td>
<td>39.2 0.44</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30.8 2.29</td>
<td>1</td>
<td>57.3 3.03</td>
<td></td>
</tr>
</tbody>
</table>

(b) LR - Cutoff: 2

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1</td>
<td></td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>Pred.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>66.0 1.17</td>
<td>0</td>
<td>24.8 0.19</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30.5 2.30</td>
<td>1</td>
<td>71.8 3.28</td>
<td></td>
</tr>
</tbody>
</table>

(d) MLP - Cutoff: 2

<table>
<thead>
<tr>
<th></th>
<th>Observed</th>
<th></th>
<th>Observed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1</td>
<td></td>
<td>0 1</td>
<td></td>
</tr>
<tr>
<td>Pred.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>66.0 1.17</td>
<td>0</td>
<td>24.8 0.19</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30.5 2.30</td>
<td>1</td>
<td>71.8 3.28</td>
<td></td>
</tr>
</tbody>
</table>

(f) RF - Cutoff: 30
**Tables**

Table 11: Minimum value, maximum value, mean and standard deviation for some of the variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthorizeAmount</td>
<td>1</td>
<td>44797</td>
<td>733.44</td>
<td>928.33</td>
</tr>
<tr>
<td>MaxLimitReturned</td>
<td>0</td>
<td>60000</td>
<td>19585.02</td>
<td>144335.97</td>
</tr>
<tr>
<td>GrandTotalOutstandingDebt</td>
<td>-3975</td>
<td>35508</td>
<td>430.54</td>
<td>698.94</td>
</tr>
<tr>
<td>RatioCreditPayment</td>
<td>0</td>
<td>4</td>
<td>0.12</td>
<td>0.22</td>
</tr>
<tr>
<td>RatioCreditIncome</td>
<td>0</td>
<td>1200</td>
<td>89.20</td>
<td>210</td>
</tr>
<tr>
<td>Income</td>
<td>0</td>
<td>1999876</td>
<td>27578.53</td>
<td>18601.91</td>
</tr>
<tr>
<td>CustomerAge</td>
<td>18</td>
<td>99</td>
<td>42.22</td>
<td>11.78</td>
</tr>
<tr>
<td>PrevDef</td>
<td>0</td>
<td>14</td>
<td>0.09</td>
<td>0.40</td>
</tr>
<tr>
<td>PrevPay</td>
<td>0</td>
<td>83</td>
<td>5.16</td>
<td>9.02</td>
</tr>
</tbody>
</table>
Figures

Figure 6: Distribution of the variable $CustomerAge$ with distribution of defaults as dark grey.

Figure 7: Distribution of the variable $MaxLimitReturned$ with distribution of defaults as dark grey.
Figure 8: Distribution of the variable *Income* with distribution of defaults as dark grey.

Figure 9: Distribution of the variable *Time* with distribution of defaults as dark grey.
Figure 10: Distribution of the variable *InternalDataExists* with distribution of defaults as dark grey.

Figure 11: Distribution of the variable *PreviousRemindersSent* with distribution of defaults as dark grey.
Figure 12: Distribution of the variable $City$ with distribution of defaults as dark grey.

Figure 13: Distribution of the variable $Seller1$ with distribution of defaults as dark grey.
Figure 14: Relative variable importance, computed with Garson’s algorithm, for the classification with the Multilayer perceptron model for the test data and partition $P_1$. 
Figure 15: ROC-curve for the classifications with Logistic regression and Multilayer Perceptron for the test data and partition $P_1$.

Figure 16: ROC-curve for the classifications with Logistic regression and Random forests for the test data and partition $P_1$. 

44