Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Håkon Meyer

An Application of Statistical Learning in Direct Marketing Response Modelling

Master's thesis in Industrial Mathematics
Supervisor: John Sølve Tyssedal
December 2019

NTNU
Norwegian University of
Science and Technology

Håkon Meyer

An Application of Statistical Learning in Direct Marketing Response Modelling

Master's thesis in Industrial Mathematics
Supervisor: John Sølve Tyssedal
December 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Preface

This thesis is submitted as a part of NTNU's master programme Industrial Mathematics. The modelling problem and the data were provided by Sparebank 1 SMN. The work has been carried out in the autumn of 2019 at the Department of Mathematical Sciences.

The thesis aims to contribute to the field of response modelling for direct marketing. It is assumed that the reader has a basic understanding of statistical modelling and is familiar with some banking terminology.

I would like to thank my main supervisor John Sølve Tyssedal and my external supervisor Jens Morten Nilsen for their counseling. Additionally, I would like to thank Sparebank 1 SMN for providing me with the opportunity to write this thesis.

Abstract

Direct marketing offers a direct means of communication between companies and prospective customers. Selecting the right target group is crucial in order to obtain the desired response, therefore response modelling is a key component in direct marketing endeavours.

With the amount of data collected and the wide variety of possible modelling methods, one can find novel and meaningful connections between the response and the explanatory variables.

The basis for this thesis is the data from a marketing campaign conducted by Sparebank1 SMN, where their clients were offered credit increases on their credit cards. The data, collected from the campaign periods stretching from March of 2015 to January of 2019, includes personal data, account data and data on spending and transactions.

Rather than performing a binary classification of the individuals, the three models employed in this thesis are used to produce a ranking of the individuals according to their willingness to respond. The estimated probability of response is ordered to produce a ranking of the individuals. The logit model, random forests and gradient boosting machines were used to estimate the probability of response.

This thesis aims to contribute by exploring how statistical learning methods can be tuned or modified to increase predictive performance, and by exploring the model's interpretation tools to better understand the relation between the explanatory variables and the response.

Sammendrag

Direkte markedsføring gir selskap en direkte form for kommunikasjon med deres kunder. For å oppnå den ønskede responsen, er det viktig å velge en passende målgruppe, derfor er responsmodellering en viktig komponent i direkte markedsføring. Med mengden data som samles, samt det store utvalget av modelleringsmetoder, kan en oppdage nye og meningsfulle sammenhenger mellom responsen og forklaringsvariablene.

Grunnlaget for denne oppgaven er data fra en markedsføringskampanje utført av Sparebank 1 SMN, der deres kunder ble tilbudt kredittøkninger på deres kredittkort. Dataene, samlet fra kampanjer utført mellom mars 2015 og januar 2019, inneholder persondata, kontodata og data knyttet til forbruk og transaksjoner.

Istedenfor å utføre en bineær klassifisering av individene, er tre modeller brukt for å rangere individer etter deres estimerte villighet til å respondere. Logit-modellen, random forests og gradient boosting machines ble brukt for å estimere sannsynlighet for respons, som kan brukes for å produsere den ønskede rangering.

Denne oppgaven har som mål å bidra ved å utforske hvordan statistiske metoder og modeller kan justeres og modifiseres for å forbedre modellenes prediksjonsevner, samt ved å utforske modellenes verktøy for å forstå sammenhenger mellom responsen og forklaringsvariablene bedre.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

It is common for companies to promote their products and services through direct marketing campaigns. Direct marketing campaigns, as opposed to mass marketing campaigns, don't promote products or services indiscriminately, but usually employ some form of data analysis to pick the target group.

There are multiple channels available with which one can conduct a direct marketing campaign. Companies can reach their target group via phone, e-mail, text messages or mail to, name a few. The different channels have different costs and benefits. Sending an e-mail to the target comes at a relatively small cost to the company, but can be easily overlooked. Conducting a direct marketing campaign by calling each individual in the target group, could result in a better response rate, but it comes at a higher cost compared to for example sending an e-mail.

Successful direct marketing campaigns can be highly profitable for the company responsible. In fact, Baesens et al. (2002) found that even small increases in the rate of response can generate large profits. There is, however, a cost associated with promoting products or services to customers. Some customers can start to feel resentment towards the company if they feel that the amount or type of offers is inappropriate. Therefore companies tend to want to target those customers who they believe would be receptive to the product or service that is offered. Modelling the customer's response can be helpful to this end.

A common practice in direct marketing response modelling is to use models to rank the prospective customers according to the estimated likelihood of response, i.e. to rank customers from likely to respond, to unlikely to respond [Berry and Linoff (2004)]. The ranking can then be used to select only the top ranked individuals as the target group for the campaign, often with the aid of a cost-benefit analysis. This allows companies to only

target the individuals where the expected profit is higher than the cost.

Direct marketing campaigns are subject to some restrictions and limitations that companies must abide by. *Markedsføringsloven* is a Norwegian law that states how marketing, and by extension, direct marketing, ought to be conducted [Norske lover (2009)]. In particular, the law specifies that people can declare that they do not wish to be contacted on certain channels such as by phone or by mail. Furthermore, privacy-related issues, such as what type of data companies can use in direct marketing response modelling, are addressed by the newly implemented EU regulations called GDPR [GDPR (2016)].

1.1 Literature Review

A multitude of different methods have been employed to model the response to direct marketing. Miguéis et al. (2017) explored methods for imbalanced data classification. Random forests in combination with undersampling outperformed other methods employed. The chosen evaluation criteria were the area under the receiver operating characteristics curve (AUC), and the later to be introduced metrics called 10% top lift and 20% top lift, which measure how well the model ranks the prospective customers in the top 10% and 20% quantiles respectively.

Ling and Li (1998) used lift exclusively to evaluate the performance of different prediction models for direct marketing response. The motivation for using lift was that is was more appropriate for direct marketing models than other metrics such as the AUC. Naive Bayes and C4.5 were the chosen methods to produce probability estimates.

Coussement et al. (2015) employed common classification techniques on four direct marketing data sets to benchmark the predictive performance. They found that some of the less interpretable prediction models, such as neural networks, performed better than traditional classifiers like logistic regression. The chosen metric for evaluation was the AUC.

1.2 Aim

The aim of this thesis is to do direct marketing response modelling for a campaign carried out by the bank Sparebank 1 SMN, which offered its customers to increase the limit on their credit cards. The campaign was conducted via e-mail between the years 2015 and 2019. The goal is to produce a prediction model that will rank prospective customers according to their willingness to respond. In addition to producing a prediction model, it is the aim of this thesis to explore the predictive power of the different methods and to examine the extent to which the models can be modified or tuned to produce better-performing prediction models. Lastly, it is the aim of this thesis to analyze the models

using available tools to better understand the relation between the explanatory variables and the response.

The data available for modelling include personal data, i.e. age and gender, data on spending and transactions, and data related to the accounts, for example credit limit, credit score, number of overdrafts, to name a few.

1.3 The Chosen Approach

The chosen approach entails creating prediction models using logistic regression, random forests and gradient boosting machines. Although random forests and gradient boosting machines both employ decision trees, their apparent resemblance is at best superficial [Friedman et al. (2001)]. Therefore, the three methods employed represent three fundamentally different methods of modelling.

Chapter 2 presents the data set and includes visualizations of correlations and distributions. Chapter 3 presents the theoretical groundwork for the methods employed, and contains some of the considerations that are typical in classification. Chapter 4 presents the results obtained from fitting and tuning the models, as well as the results from the attempts to interpret the models. In chapter 5 the results are discussed and some considerations and recommendations for future work are presented.

Chapter 2

The Data Set

The data set provided by Sparebank 1 SMN consists of 627113 observations and 72 variables. Each observation represents a credit card user who has received an offer to apply for a credit limit increase. The offers were sent out between March of 2015 and January of 2019. The flagging variable *ResponseInd2* denotes whether the customer has decided to apply for an increase.

2.1 Variables

A full list of variables with a short description can be seen in the appendix. Information related directly to the person is limited to only age and gender. Information such as marital status, occupation, salary and home ownership, is not available for this modelling project. There are several variables related to the account and spending habits. These variables include, but are not limited to, the credit limit, the name of the bank, the number of days before the credit card is used, and the closing balance. Some variables have been aggregated over different periods, usually three and twelve months prior to time the offer is sent out.

2.2 Response

Customers have received a total of 627113 offers from their bank and customers have chosen to apply for an increase 84872 times, which means that the response rate for the whole data set is 0.1353%. Table 2.1 shows the distribution.

	Respondent	Non-Respondent
Number	84872	542241
Proportion	0.1353	0.8647

Table 2.1: Distribution of the response.

The response variable *ResponseInd2* is a flagging variable which denotes whether a customer has chosen to act on the offer and thus applied for a credit increase. *ResponseInd2* is 1 for customers who've chosen to apply and 0 for those who did not apply. For the latter case, it is not possible to tell if the customer did not register the offer, or if he or she simply was not interested.

Not all of the customers who apply, are granted an increase, in fact, 5992 of the 84872 applications were declined. It may seem odd that some customers who were the target of the campaign had their applications declined, but the reason for this is that the customers must report information which was previously unavailable, and the bank reserves the right to decline on the basis of this new information. Although it could be interesting to model which applications would be declined, this task is not within the scope of this thesis.

2.3 Visualizing the Data

A correlation plot of the continuous variables can be seen in Figure 2.1. Some variables are the same quantity aggregated over different periods, such as *SumAirlineL3* and *SumAirlineL12*, denoting the sum paid to airlines the preceding 3 and 12 months, respectively. These variables are clearly correlated on the plot. In fact, there are 15 such variables related to spending, that are aggregated over 3 and 12 months. Their correlation is marked by the 15-element long diagonal line of blue dots off the correlation plot's diagonal.

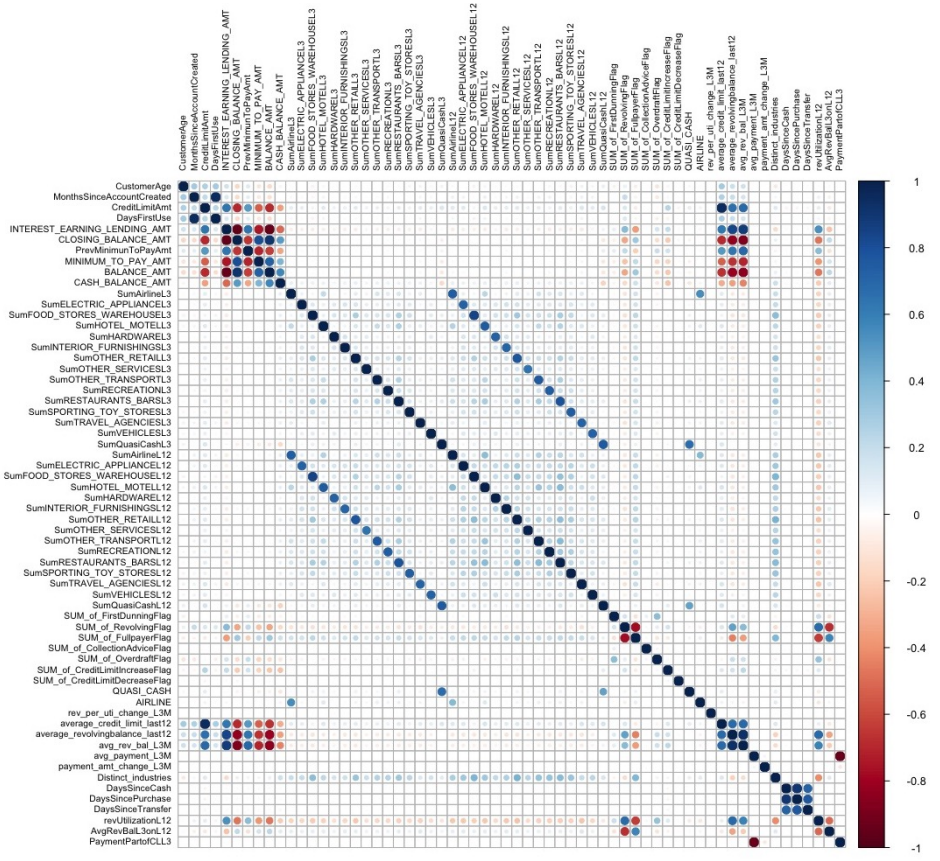


Figure 2.1: Correlation plot of the continuous variables.

Moreover, the variable *DaysFirstUse*, denoting the number of days before the credit card is used, is highly correlated with the *MonthsSinceAccountCreated*, denoting the number of months since the account has been created. This means that customers of newer accounts have been more eager to use their credit cards than customers that have been with the bank longer.

2.3.1 Visualizing the Response Rate for Individual Variables

The response rate can be plotted against individual variables to show how the average response rate varies for different values within the variable. That way one can see the distribution of response for individual variables. Sometimes these plots can be misleading if the groups contain a small number of observations. For example, a group may have an unusually high response rate, but if the number of customers belonging to this group is

very low, it may not warrant special attention. Therefore, the relative distribution of the groups is included in the plots, marked by the grey bars. Visualizing the distribution of the groups can also be helpful, regardless of the response rate, as a means of getting to know the data.

The average response rate for different credit limits can be seen in Figure 2.2. The blue lines and blue dots represent the response rate and the grey opaque bars show the relative frequency of the groups. The figure shows that the lowest response rate is registered for customers with a credit limit below 10000 kroner and the highest response rate is registered for customers with a credit limit of more than 50000 kroner. In fact, for the latter group, the response rate is over 0.35, which is remarkably high. Around 13.5% of customers belong to this latter group, as indicated by the gray bar.

Figure 2.3 shows the account balance as a proportion of the credit limit. If that quantity is lower than 0, then that customer has a positive balance on his or her credit card. The response rate is high for those customers who have a large balance to limit ratio, and the rate is particularly high for customers whose balance to limit ratio is over 1. These individuals have exceeded their credit limit and are seemingly particularly eager to increase their limits. Their response rate is over 0.2, and they make up around 3% of the customers, as indicated by the grey bar.

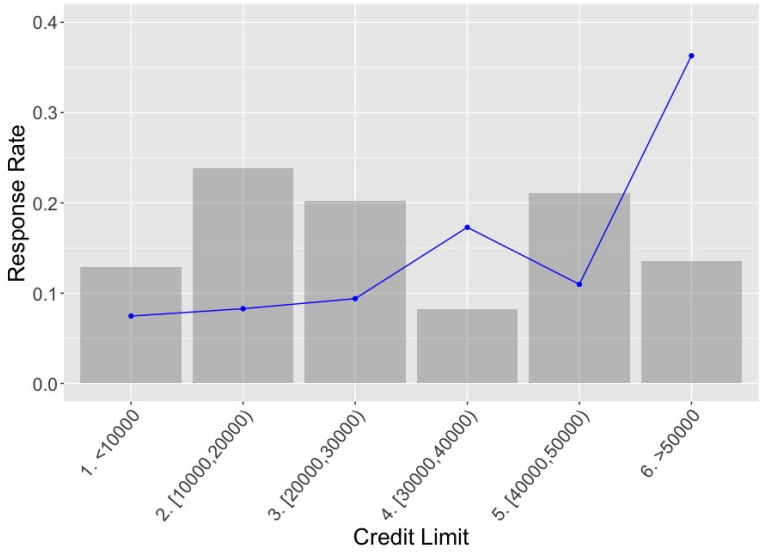


Figure 2.2: Response rate for different groups of the credit limit.

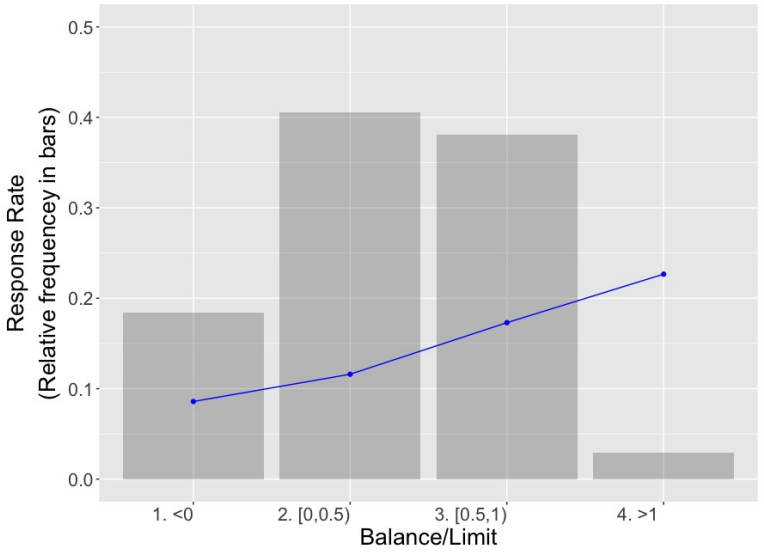


Figure 2.3: Response rate for different groups of the balance to credit limit ratio.

The response rate for different time periods can be seen in Figure 2.4. From the figure, one can see that the response rate has been declining. In the first half of 2015 the average response rate was around 0.2. Then, in the period between the end of 2015 and the start of 2018, the response rate has fluctuated around the 0.10 mark. And finally, in the second half of 2018 and the start of 2019, the recorded response rate was closer to 0.05. On the basis of this plot, it seems probable that the future response rate will be around 0.05, assuming the that same type of customers are targeted.

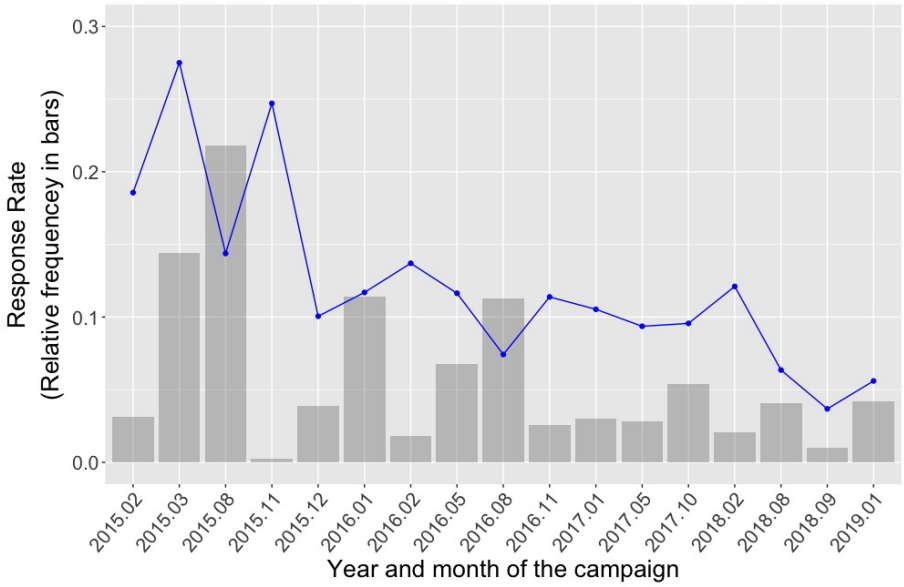


Figure 2.4: Response rate for different campaign periods.

Figure 2.5 shows the response rate for different groups of the variable *MonthsSinceAccountCreated*, denoting the number of months since the account was created. The response rate is smaller for larger number of months, i.e. the response rate is higher for newer accounts than for older accounts. In fact, for accounts created less than 25 months prior to the campaign, the response rate is close to 0.2 and for accounts created more than 125 months prior to the campaign, the response rate is below 0.1.

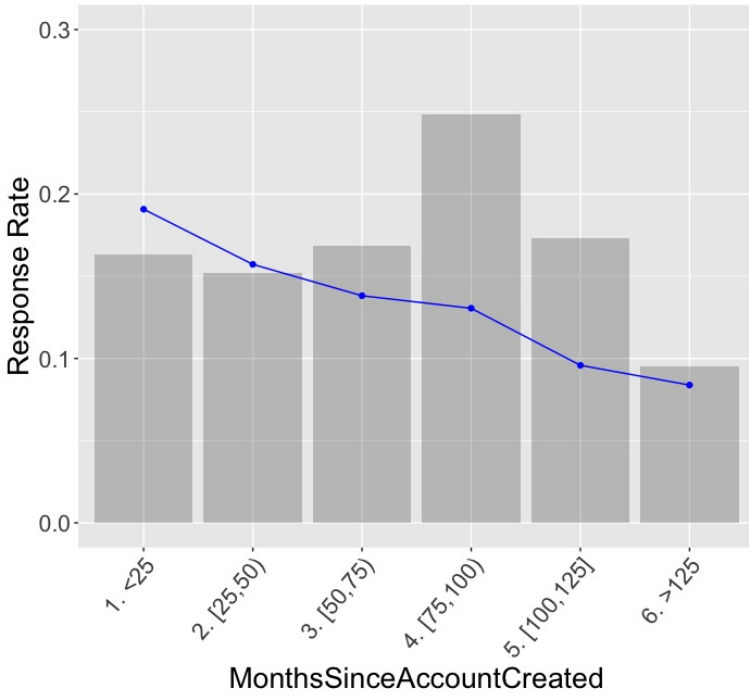


Figure 2.5: Response rate for the variable *MonthsSinceAccountCreated*, denoting the number of months since the account was created.

Figure 2.6 shows the response rate for different groups of the variable *DaysFirstUse*, denoting the number of days before the credit card is used. The response rate is smaller for larger number of days, meaning that customers who use the card faster, are on average more willing to respond to the campaign, than those customers who are slower in this respect.

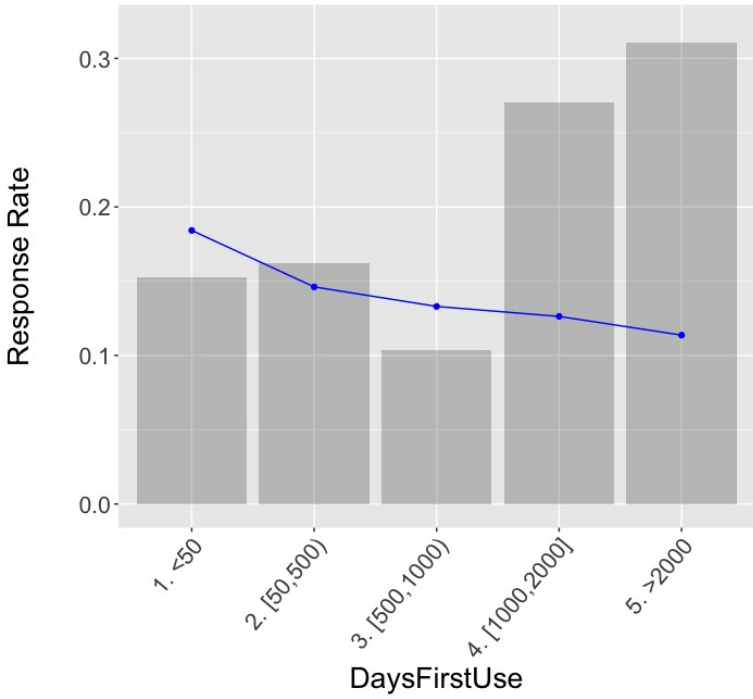


Figure 2.6: Response rate for the variable *DaysFirstUse*, denoting the number of days before the credit card is used.

The response rate for different credit scores can be seen in Figure 2.7. Lower credit scores are thought to be better, i.e. individuals with low scores are thought to be more creditworthy. The plot suggests that customers with high credit scores are more inclined to respond to the campaign.

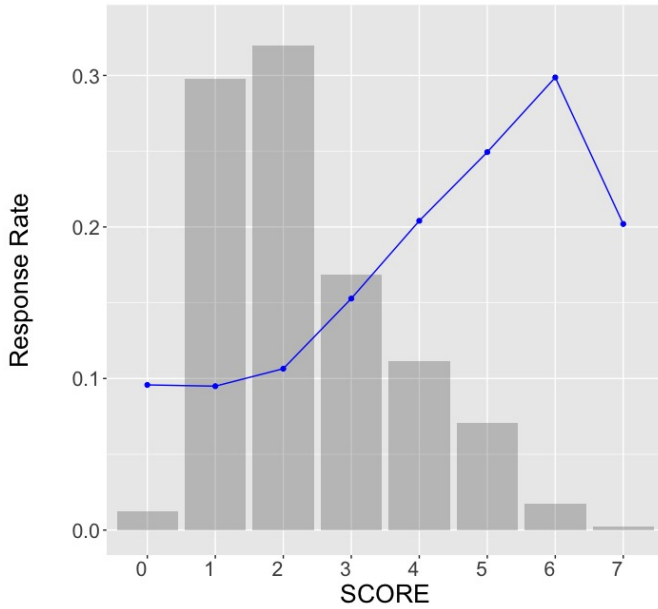


Figure 2.7: Response rate for different credit scores

Chapter 3

Theory

In this chapter, the theoretical framework for the models is presented. Additionally, some performance metrics are introduced and some other statistical tools are presented.

3.1 Binary Regression and the Logit Model

Let the response vector be denoted by $\boldsymbol{Y}$ and let the ith response value be denoted by y_i. Furthermore, let the $(n \times p)$-design matrix be denoted by $\boldsymbol{X}$, where n is the number of data points and p is the number of covariates including the intercept term. Let the ith data point be denoted by $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ and let $k = p - 1$ be the number of covariates.

$$\eta_i = \boldsymbol{x}_i^T \boldsymbol{\beta} = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{ik}\beta_k \, .$$

Assuming the response y_i can only take values 0 or 1 and that y_i takes the value 1 with probability p_i, then the response y_i is said to have a Bernoulli distribution, that is $y_i \sim B(1, p_i)$, and its discrete probability density function is given by

$$f(y_i \mid \pi_i) = \pi_i^{y_i}(1 - \pi_i)^{1-y_i} \, .$$

Binary regression aims to model the conditional probability of y_i being 1, denoted by

$$\pi_i = P(y_i = 1) = E(y_i) \, ,$$

given the covariate values $\boldsymbol{x}_i$. The effects of the explanatory variables are modelled through a linear predictor $\eta_i = \boldsymbol{x}_i^T \boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ are the p regression coefficients to be estimated. The linear predictor can take all real values, thus to ensure

that the estimated probability π_i lies in the interval $[0, 1]$, the function that links the linear predictor to the probability π_i must be a cumulative distribution function defined on all real values. Assuming h is such a function, we have that the estimated probability π_i is linked to the linear predictor by:

$$\pi_i = h(\eta_i) \, .$$

$h(\eta)$ is referred to as the response function, and its inverse $h^{-1} = g$ is referred to as the link function.

The logistic response function is a common choice for h. It is given by

$$\pi_i = h(\eta_i) = \frac{\exp \eta_i}{1 + \exp \eta_i} \, . \tag{3.1}$$

Binary regression with this response function is called logistic regression, and the model obtained by doing logistic regression is called the logit model. The link function for the logit model is given by

$$\eta_i = \log \left(\frac{\pi_i}{1 - \pi_i} \right) = \log \left(\frac{P(y_i = 1)}{1 - P(y_i = 1)} \right) . \tag{3.2}$$

Taking the exponent on both sides of Equation 3.2, we get the ratio

$$\frac{P(y_i = 1)}{P(y_i = 0)} = \exp \left(\beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \cdots + x_{ik}\beta_k \right) . \tag{3.3}$$

This ratio is usually referred to as the odds. If the odds are $\frac{1}{2}$, then the probability of y_i belonging to either class is equally likely. If the odds are $\frac{1}{3}$, then the probability of y_i belonging to class 0 is three times as high as the probability of y_i belonging to class 1. In other words, if the odds are $\frac{1}{3}$, then $P(y_i = 1) = 25\%$ and $P(y_i = 0) = 75\%$. It is worth noting that the odds do not have an upper limit, but do have a lower limit of 0.

3.1.1 Parameter Interpretation for the Logit Model

Parameter interpretation for the logit model usually involves looking at the effect the covariates have on the odds. We can rewrite Equation 3.3 by noting that the exponent of a sum of elements is equal to the product of the exponents of each element, i.e.

$$\frac{P(y_i = 1)}{P(y_i = 0)} = \exp \beta_0 \cdot \exp x_{i1}\beta_1 \cdot \exp x_{i2}\beta_2 \cdots \exp x_{ik}\beta_k \, .$$

Assume the value of a covariate x_{ij} increases by 1, then the new odds will be

$$\frac{P(y_i = 1)}{P(y_i = 0)} = \exp \beta_0 \cdot \exp x_{i1}\beta_1 \cdot \exp x_{i2}\beta_2 \cdots \exp (x_{ij} + 1)\beta_j \cdots \exp x_{ik}\beta_k \ ,$$

which is equal to

$$\frac{P(y_i = 1)}{P(y_i = 0)} = \exp \beta_j \cdot [\exp \beta_0 \cdot \exp x_{i1}\beta_1 \exp x_{i2}\beta_2 \cdots \exp x_{ij}\beta_j \cdots \exp x_{ik}\beta_k] \ . \quad (3.4)$$

The odds in Equation 3.4 are scaled by a factor of $\exp \beta_j$ when the value of the jth covariate increases by 1. To assess the effect of a change in the value of the jth covariate, we consider three different cases for the value of the coefficient estimate β_j. If $\beta_j > 0$, then an increase in jth covariate of 1, results in an increase in the odds by a factor of $\exp(\beta_j)$. Similarly, if $\beta_j < 0$, then an increase in the value of the jth covariate of 1, results in an decrease in the odds by a factor of $\exp(\beta_j)$. If $\beta_j = 0$, then a change in the jth covariate does not affect the odds.

Note that, as opposed to a multivariate linear regression model, the change in the model response π_i depends on the current value of $\mathbf{x}_i$. Figure 3.1 illustrates this. The figure

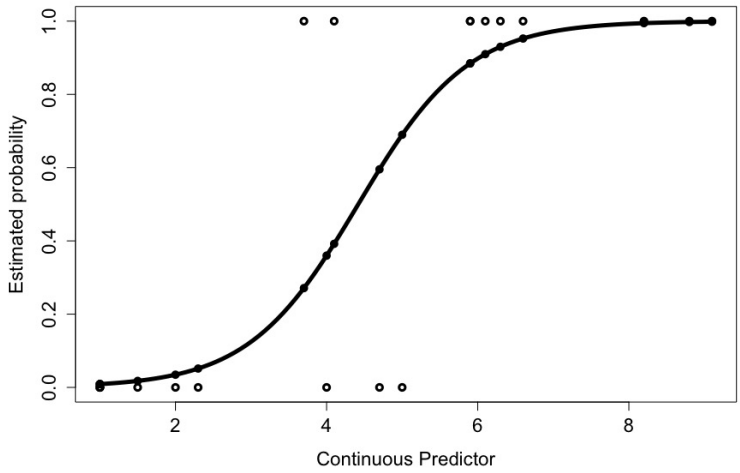


Figure 3.1: Illustration of a logit model with a single predictor.

shows a typical logit model with a single continuous predictor. The relationship between the estimated probability of response and the continuous predictor is given by an S-shaped curve, which means that the change in the estimated probability per unit change in the predictor is dependent on the current value of the predictor. To illustrate this, consider the

case where the estimated probability is already very high or very low, then an increase in the predictor will result in a relatively small change in the estimated probability.

3.1.2 Parameter Estimation Using Maximum Likelihood

Maximum likelihood (ML) estimation is the most common way to find parameter estimates. Assuming the responses are conditionally independent, the likelihood can be written as $L(\boldsymbol{\beta})$ and is given by

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\beta}) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} ,$$

where $y_i = 0, 1$.

The ML estimates $\hat{\boldsymbol{\beta}}$ are the values for $\boldsymbol{\beta}$ that maximize the likelihood $L(\boldsymbol{\beta})$. Maximizing the log of the likelihood gives the same estimates and is often more convenient to work with. The log likelihood is given by

$$\begin{aligned} l(\boldsymbol{\beta}) &= \sum_{i=1}^{n} l_i(\boldsymbol{\beta}) \\ &= \sum_{i=1}^{n} y_i \log(\pi_i) - y_i \log(1-\pi_i) + \log(1-\pi_i) \\ &= \sum_{i=1}^{n} y_i \log(\frac{\pi_i}{1-\pi_i}) + \log(1-\pi_i) . \end{aligned}$$

From Equation 3.2, we have that $\boldsymbol{x}_i^T\boldsymbol{\beta} = \eta_i = \log\left(\frac{\pi_i}{1-\pi_i}\right)$ for the logit model. Furthermore, it can be shown that $(1-\pi_i) = (1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}))^{-1}$, which yields

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \boldsymbol{x}_i^T\boldsymbol{\beta} - \log(1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta})) .$$

Now we have an expression for $l_i(\boldsymbol{\beta})$ that we can differentiate with respect to $\boldsymbol{\beta}$. We get

$$\frac{\partial l_i(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \boldsymbol{x}_i(y_i - \pi_i) .$$

For convenience, let us introduce a *Score Function*

$$S(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \boldsymbol{x}_i(y_i - \pi_i) .$$

S($\boldsymbol{\beta}$) is a vector of length p with $\frac{\partial l(\boldsymbol{\beta_i})}{\partial \boldsymbol{\beta_i}}$ as its ith element, for $i = 0, 1, \ldots, k$

The ML estimates can be obtained by setting the score function to zero, i.e.

$$S(\hat{\boldsymbol{\beta}}) = 0 \tag{3.5}$$

Solutions to Equation 3.5 are usually found iteratively by either the Newton-Raphson algorithm or the Fisher scoring algorithm [Fahrmeir et al. (2013)]. The Newton-Raphson method makes use of the negative Hessian of $l(\boldsymbol{\beta})$, often referred to as the observed information matrix $H(\boldsymbol{\beta})$, and the Fischer scoring algorithm makes use of the expected information matrix $F(\boldsymbol{\beta}) = E[H(\boldsymbol{\beta})]$.

The (i, j)th matrix element of the observed information matrix is given by

$$H_{ij}(\boldsymbol{\beta}) = -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_i \partial \boldsymbol{\beta}_j^T} ,$$

which can be written more compactly as

$$H(\boldsymbol{\beta}) = -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} .$$

The expected information matrix is given by

$$F(\boldsymbol{\beta}) = E\Big[-\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \Big] .$$

Using the Fisher scoring algorithm, the solution is found by using the iteration scheme given by

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t + \boldsymbol{F}^{-1}(\boldsymbol{\beta}^t) S(\boldsymbol{\beta}^t) ,$$

where t is the current iteration. With an initial guess $\boldsymbol{\beta}^0$, the scheme iterates until convergence. For the algorithm to converge, it is required that $\boldsymbol{F}(\boldsymbol{\beta})$ is invertible for all values of $\boldsymbol{\beta}$, this in turn requires that the design matrix has full rank. Thus, if the design matrix contains any linearly dependent columns, the iteration scheme will not converge.

For the choice of the logistic response function, we have the convenient property that the expected information matrix is equal to the observed information matrix, i.e. $F(\boldsymbol{\beta}) = H(\boldsymbol{\beta})$. Furthermore, the estimated regression coefficients $\hat{\boldsymbol{\beta}}$ are asymptotically distributed as

$$\hat{\boldsymbol{\beta}} \approx \mathrm{N}(\boldsymbol{\beta}, F^{-1}(\hat{\boldsymbol{\beta}})) .$$

That is, for sufficiently large values of n, $\hat{\boldsymbol{\beta}}$ has a multivariate normal distribution with expected value $\boldsymbol{\beta}$ and covariance matrix equal to the inverse of the observed information matrix.

3.1.3 Dummy Variable Coding

Dummy variable coding is a common way to deal with categorical variables in regression models. Assuming the jth explanatory variable has m categories, then we use $m-1$ dummy variables in our regression model and the omitted category serves as a reference category.

For a given observation x_j, a dummy variable takes the value 1, if the observation belongs to its particular category, and 0 otherwise, i.e. for category i

$$x_{ij} = \begin{cases} 1, & \text{if } x_j \text{ belongs to category } i. \\ 0, & \text{otherwise.} \end{cases}$$

The way the parameter estimates are interpreted when using dummy variable coding, is by comparing them to the reference category. Although one can choose any category to serve as the reference category, it is common for the sake of interpretation, to pick the category that occurs most frequently.

3.1.4 Stepwise Variable Selection

In order to avoid including irrelevant variables in the regression model, some form of variable selection is often warranted. Ideally one would want to test every possible combination of predictors to obtain the best model, but this can be computationally intensive and is often not feasible with a large number of candidate variables. Stepwise methods represent a more computationally efficient method of doing variable selection. Backwards selection is an example of a stepwise method. It entails initially fitting all variables, and then iteratively removing variables according to a chosen criterion. The Bayesian Information Criterion (BIC), introduced by Schwarz (1978), can serve as the selection criterion. It is defined as

$$BIC = k\ln(n) - 2\ln(\hat{L}) \,,$$

where $\hat{L}$ is the maximized likelihood function. At each step of the elimination procedure, the variable that corresponds to the largest *decrease* in the BIC, is eliminated. The procedure stops when the BIC cannot be reduced further by omitting a variable.

3.1.5 L_1 Penalty for Logistic Regression

Introducing a penalty term when fitting a logit model is a way to apply shrinkage, i.e. to constrain the parameter estimates. The motivation for using shrinkage is that it can result in a model with lower variance at the cost of a small increase in bias. One of key properties of L_1 penalty term, when applied to a linear regression model, is that it can be

used to perform variable selection, due to the fact that it shrinks parameter estimates to zero for finite values of the penalty hyperparameter.

Recall from Equation 3.1.2, that the log-likelihood for the logit model is

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \log(1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})) \, .$$

We can introduce shrinkage to the logit model by adding a penalty term to the log likelihood. The regularized logit model parameter estimates are found by:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left[\sum_{i=1}^{n} y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \log(1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta}) - \lambda \sum_{j}^{p} |\beta_j| \right] \, , \tag{3.6}$$

where λ is the penalty hyperparameter. Typically the variables are standardized in order for the penalty term to make sense. Standardizing the variables means that the intercept term is adjusted, but intercept term is usually not penalized [Friedman et al. (2001)].

If we let the penalty hyperparameter $\lambda \to \infty$, then $\boldsymbol{\beta} \to 0$. Moreover, when $\lambda = 0$, the parameter estimates obtained are the same as those obtained when fitting a non-penalized logit model.

The maximization problem in Equation 3.6 is concave. There are different methods available in order to find the solution. The common R package *glmnet* uses cyclical coordinate descent to find the solution, which entails optimizing the objective function successively for each parameter while the others are fixed [Friedman et al. (2010)]. The algorithm repeats this procedure several times until convergence.

3.2 Random Forest and Tree-based Methods

Whereas regression models methods seek to model the effect of predictors on the response, a tree-based method entails segmenting the predictor space into non-overlapping regions. They represent fundamentally different approaches to creating prediction models.

3.2.1 Constructing a Classification Tree

Assume we have k predictors $X_1, X_2, \ldots X_k$, then the construction of a regression tree consists of finding splits to divide the predictor space into L non-overlapping regions $R_1, R_2, \ldots R_L$. A representative example of a classification tree can be seen in Figure 3.2. The algorithm to construct a classification tree works by utilizing recursive binary splitting to produce two new terminal nodes until a stopping criteria is met. The stopping criteria can for example be a maximum interaction depth d, which is equal to the number

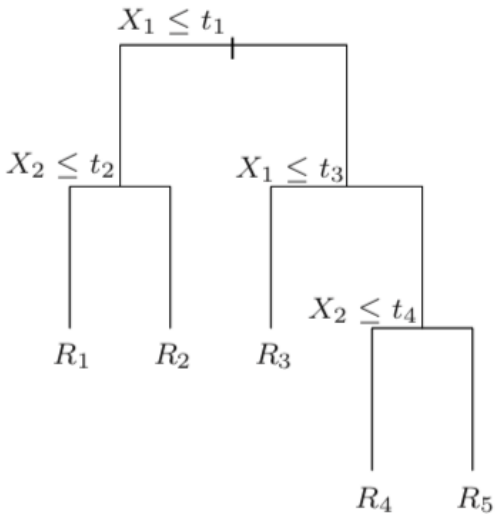


Figure 3.2: Example of a decision tree.

of splits made. A minimum number of training examples in a terminal node, n_{min}, can also serve as a stopping criteria. The most common splitting rule for classification trees utilizes the Gini index to determine the split. For a given region R_l, the Gini index is defined as

$$G = \sum_{i=1}^{2} p_{il}(1 - p_{il}) \,, \tag{3.7}$$

where p_{il} is the proportion of observations in the lth region that belong to the ith class. From Equation 3.7 one can see that the gini index for a given region R_l is low when p_{il} is close to 0 or 1, for the two possible classes $i = 0$ and $i = 1$. So, the gini index is low when observations in a region belong mainly to a single class, i.e. when node purity is high. The split that reduces the Gini index the most, is chosen among the possible splits. As we can tell from Figure 3.2, each terminal node corresponds to a region. Assuming a classification tree is trained and assuming that an observation x_i belongs to the region R_l of the tree, then the prediction $\hat{f}(x_i)$ assigned to observation x_i, is equal to the class that occurs the most in R_l. For example, consider a two-class classification tree. If the class that occurs most often in region R_l during training is class 1, then the classification tree assigns the prediction 1 to the region R_l. Any new observation x_{new} belonging to the region R_l will be assigned the prediction 1.

3.2.2 Bagging and Random Forests

A single classification tree is easy to visualize and interpret, but its predictive performance, due to its high variance, is often lacking when compared to other common prediction models. It does however lay the foundation for creating better-performing prediction models. By Bootstrap Aggregating (bagging) decision trees, one can reduce the variance considerably. The idea was proposed by Leo Breimann [Breiman (1996)] and it entails fitting multiple decision trees using bootstrapped training sets, and averaging their predictions. For every decision tree, only a proportion (typically $\frac{2}{3}$) of the total available training data is used in constructing the tree.

Assume B trees are trained, each from its own bootstrapped sample of the training data, resulting in B decision trees $\hat{f}_1(x), \hat{f}_2(x), \ldots, \hat{f}_B(x)$. The final model is then the average of the B models,

$$\hat{f}(x) = \frac{1}{B} \sum_{i}^{B} \hat{f}_i(x) \, .$$

If bagging is applied to perform classification, then each individual tree makes a prediction as to what class an observation belongs to. If one wants the model to assign a class prediction to each training observation, the majority vote can be used, i.e. the final prediction is equal to the most commonly occurring class. It is also possible to use the *proportion* of class occurrences, to produce a probability estimate for the observation belonging to a particular class. For example, in a two-class classification problem, a model might have 80 trees predicting that a particular observation belongs to class 1 and 20 trees predicting that it belongs to class 0. The model can then assign the prediction $\frac{80}{80+20} = 0.8$ to the observation, representing the model's confidence in the observation belonging to class 1.

A drawback with bagged trees is that the trees are still correlated. By using a small modification, however, one can address this issue. *Random Forests* [Breiman (2001)] is similar to bagging, but for each split, only a random sample of the p total predictors are considered. A common choice is to train with a randomly selected $m = \sqrt{p}$ predictors for each split. This modification helps to decorrelate the trees and therefore represents an improvement over the standard decision tree bagging.

To illustrate the added benefit of random forests over standard bagging trees, we can imagine that there is predictor which is able to explain the variability in the data much better than the other predictors. If we allow each tree to utilize all predictors at each split, then the stronger predictor will be the first split in all the trees, and the trees will be rather similar to each other. This leads to a higher variance in the prediction. Random forests only consider a random subset of the total predictors for each split, thus ensuring that the trees are less correlated and thus producing predictions with less variance.

Let the number of trees be denoted by B and let the minimum node size be denoted by

n_{min}, then basic procedure for training a random forest is given by:

1. For b=1 to B
   (a) Produce a bootstrap sample from the training data
   (b) Recursively perform the following steps for all terminal nodes until n_{min} is reached:
      i. Sample m of the possible predictors
      ii. Apply a split on the predictor, among the m possible predictors, that reduces the Gini index the most and produce two new terminal nodes
2. Average the B trees to produce the final model

3.2.3 Hyperparameter Tuning

Random forests have many hyperparameters that are possible to tune. The most commonly tuned hyperparameters are the number of trees B and the number of sampled predictors m. It is also possible to consider different values for the minimum number of observations per node n_{min} in training.

Random forests do not overfit with increasing number of trees [Breiman (2001)]. So, selecting a high number of trees and varying the other hyperparameters is one possible strategy.

There is no reason to assume that the hyperparameters are independent of each other with respect to the predictive performance, therefore tuning them individually is not an optimal strategy. One possible strategy is to train with a number of different combinations of hyperparameters in a cross-validation experiment and select the hyperparameters for which the model obtains the best average performance on the left-out validation sets.

3.3 Gradient Boosting Machines

Boosting is a powerful method that entails training several weak classifiers and combining them. A weak classifier in this context is a classifier that performs only slightly better than randomly guessing. Although boosting can be applied using several different base learners, it is with the decision tree as base learner that one may see some of the most substantial improvements [Friedman et al. (2001)]. *Gradient boosting machines* (GBM) is a method that uses decision trees as the base learner, and it is thought to be robust and to have competitive performance [Friedman (2001)].

Gradient boosting works by fitting regression trees to the residuals of the preceding trees. In order to apply the gradient boosting algorithm, one must choose a differentiable loss function $L(y_i, f(\mathbf{x}_i))$, where $f(\mathbf{x}_i)$ is the log-odds of observation $\mathbf{x}_i$ belonging to class 1. Recall from the logit model that the log-odds for the observation $\mathbf{x}_i$ are defined as

$$\log\left(\frac{P(y_i = 1)}{P(y_i = 0)}\right).$$

The algorithm for training a gradient boosting machine to produce class probabilities is given by:

(1) Set hyperparameters: Number of trees B, interaction depth d, learning rate λ and the minimum node size n_{min}.

(2) Initialize model with constant value $f_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \gamma)$.

(3) For b=1 to B

(3a) Compute residuals $r_{ib} = -\left[\frac{\partial L(y_i, f_{b-1}(x_i))}{\partial f_{b-1}(x_i)}\right]$ for $i = 1, \ldots, i = N$

(3b) Fit a regression tree with maximum depth d to the residuals r_{ib}, producing the regions $R_{1b}, R_{2b}, \ldots R_{Jb}$

(3c) For j=1 to J: compute $\gamma_{jb} = \underset{\gamma}{\operatorname{argmin}} \sum_{\mathbf{x}_i \in R_{ij}} L(y_i, f_{b-1}(\mathbf{x}_i) + \gamma)$

(3d) Update the response: $f_b(\mathbf{x}_i) = f_{b-1}(\mathbf{x}_i) + \lambda \sum_{j=1}^{J_b} \gamma_{jb} I(\mathbf{x}_i \in R_{jm})$ for $i = 1, \ldots, i = N$

(4) Output the B trees

In (1) the hyperparameters are set. Typical hyperparameters that warrant special attention are the number of trees, B, the interaction depth, d, the learning rate, λ, and the minimum node size, n_{min}.

In (2), the model is initialized with a tree that consists of a single terminal node, i.e. we start with the optimal constant model. This optimal constant model is the log of proportion of the training examples belonging to class 1 in the training set, i.e. if 10% of the examples belong to class 1 in the training set, then $f_0 = \log(0.1)$.

In (3) the trees are being trained sequentially. Following the computation of the residuals in (3a), a new regression tree is fitted to the residuals, producing new terminal nodes in (3b). For each of the J terminal nodes we obtain, the γ value is chosen in (3c) to be

the value for which the loss function is minimized. Then the response is updated in (3d). The magnitude of the update is determined by the learning rate λ. Since we say that the learning rate *shrinks* the contribution of each tree, the learning rate is often referred to as the shrinkage parameter. A typical value for the learning rate is 0.1.

Assuming B trees have been fitted sequentially according to this procedure, the output in (4) are the B trees, and the model output is $f(\mathbf{x}_i) = f_B(\mathbf{x}_i)$ for an observation $\mathbf{x}_i$ in the training set.

The most common choice for the loss function in classification is the binomial negative log-likelihood, which is given by

$$L(y_i, p_i) = L_i = -\left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\right].$$

This loss function is written as a function of $p_i = \Pr(y_i = 1 \mid \mathbf{x}_i)$. In order to correctly implement the gradient boosting method, we need a loss function that is differentiable with respect to the log-odds $f(\mathbf{x}_i)$, so we need to rewrite the loss function as a function of the log-odds:

$$\begin{aligned} L_i &= -\left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\right] \\ &= -y_i \log(p_i) - (1 - y_i) \log(1 - p_i) \\ &= -y_i \log(p_i) - \log(1 - p_i) + y_i \log(1 - p_i) \,. \end{aligned}$$

We use that

$$\log(p_i) - \log(1 - p_i) = \log \frac{p_i}{1 - p_i} = \log \text{odds}_i$$

and that

$$p_i = \frac{\exp(\log \text{odds}_i)}{1 + \exp(\log \text{odds}_i)}$$

to obtain

$$\begin{aligned} L_i &= -y_i[\log \text{odds}_i] - \log(1 - p_i) \\ &= -y_i[\log \text{odds}_i] - \log\left(1 - \frac{\exp(\log \text{odds}_i)}{1 + \exp(\log \text{odds}_i)}\right) \\ &= -y_i \log \text{odds}_i - \log\left(\frac{1}{1 + \exp\left(\log \text{odds}_i\right)}\right) \\ &= -y_i \log \text{odds}_i - \left[\log 1 - \log(1 + \exp\left(\log \text{odds}_i\right))\right] \\ &= -y_i \log \text{odds}_i - \left[-\log(1 + \exp\left(\log \text{odds}_i\right))\right] \\ &= -y_i \log \text{odds}_i + \log(1 + \exp\left(\log \text{odds}_i\right)) \,. \end{aligned}$$

Alternatively, we can write that

$$L(y_i, f(\mathbf{x}_i)) = -y_i f(\mathbf{x}_i) + \log(1 + \exp f(\mathbf{x}_i)) \; . \quad (3.8)$$

In Equation 3.8, the loss function is written as a function of the log-odds, which is what we wanted.

By using the binomial negative log-likelihood as the loss function, the residuals, r, are equal to the negative derivative of the loss function with respect to the log-odds, i.e.

$$\begin{aligned} r_{ib} &= -\left[\frac{\partial L(y_i, f_{b-1}(x_i))}{\partial f_{b-1}(x_i)}\right] \\ &= -\left[\frac{\partial[-y_i f_{b-1}(x_i) + \log(1 + \exp(f_{b-1}(x_i)))]}{\partial f_{b-1}(x_i)}\right] \\ &= -\left[-y_i + \frac{\exp(f_{b-1}(x_i))}{1 + \exp(f_{b-1}(x_i))}\right] \\ &= -[-y_i + p_i] \\ &= y_i - p_i \; . \end{aligned} \quad (3.9)$$

From Equation 3.9 we see that the residuals are equal to the difference between the observed class $\{1, 0\}$ and the estimated probability that the training example belongs to class 1.

Assuming a gradient boosting model is trained, the prediction for a new observation $\mathbf{x}_{new}$ is found by passing the observation through the B trees and updating $f(\mathbf{x}_{new})$ according to the learning rate, until $f_B(\mathbf{x}_{new})$ is reached. Then the probability estimate is found by:

$$p(\mathbf{x}_{new}) = \Pr(y_{new} = 1 \mid \mathbf{x}_{new}) = \frac{e^{f_B(\mathbf{x}_{new})}}{1 + e^{f_B(\mathbf{x}_{new})}}.$$

3.3.1 Hyperparameter Analysis

The learning rate is λ, which is also known as the shrinkage parameter, is an important hyperparameter. It controls the magnitude of the contribution of each tree. We can *shrink* the contribution of each tree by having a small learning rate. Moreover, the learning rate λ and the number of trees B are closely related in regards to the training risk. Smaller values of the learning rate require a larger number of trees to obtain the same training risk. Friedman (2001) argues that the learning rate ought to be low and that the number of trees should be as high as computationally feasible, as opposed to a high learning rate and a smaller number of trees, to obtain more favorable generalization performance.

The interaction depth d, is also a key component in hyperparameter analysis for gradient boosting machines. It controls the number of splits, and thus also the level of the

interaction effects. If $d = 1$, then the trees are all stumps, which means that each tree only has a single split. When all the trees are stumps, only main effects are modelled, i.e no higher-order effects are modelled when $d = 1$. With $d = 2$, the trees can have 2 splits and thus second-order interaction effects are permitted, i.e. two-variable interaction effects can also be modelled. If the low-order interaction effects dominate, then d can be low. Very high interaction depth levels ($d > 20$) is found to provide little added benefit over more compact trees [Natekin and Knoll (2013)].

3.3.2 Interpretation

As with random forest, directly interpreting the large of number of trees in gradient boosting is difficult, but there are some interpretation tools available. Calculating the relative importance of the predictors can provide us with insight into what predictors play the biggest role.

In order to calculate the relative importance of predictors, consider a predictor k, and a single regression tree T in a GBM model. Recall that the regression trees in the GBM model are fitted to the residuals of the preceding trees. Moreover, the chosen splits for each tree are the splits that reduce the squared error the most. Assuming the tree T has L terminal nodes, there are $L - 1$ splits in the tree. We can define the influence of the kth predictor in the regression tree as

$$\text{Infl}_j(T) = \sum_{i=1}^{L-1} I_i^2 \mathbb{1}(S_i = k)\,, \tag{3.10}$$

where I_i^2 is the resulting improvement in the squared error from the split and S_i is the predictor chosen in the ith split, and $\mathbb{1}(S_i = k)$ is an indicator function. In order to obtain the influence of the kth predictor on the whole model, the influence is summed over the B trees. In other words, to find the importance of a predictor, we sum the reduction in squared error over all the splits on that predictor in the trees.

The influence for all predictors is then scaled to the most influential predictor for easier comparison. The amount of influence does not, however, say anything about how the predictor affects the response. Another interpretation tool such as the partial dependence can be helpful in this regard.

Partial dependence plots (PDP) serve as a means of analyzing the effect individual predictors have on the response. For classification, PDPs give an insight into how the log-odds depend on individual predictors. Let $X = (X_1, X_2, \ldots, X_p)$ be all the predictors in a model where the response is the log-odds, denoted by $f(X)$. Now consider the partition of X into X_k, the predictor of interest, and its complement $X_c = X \setminus X_k$. Then the partial

dependence of the response $f(X)$ on predictor X_k, denoted by $f_k(X_k)$, is defined as:

$$f_k(X_k) = \mathrm{E}_{X_c}[f(X_k, X_c)] \,,$$

i.e. the partial dependence of a predictor is the marginal average of the response. Note that $f(X)$ and $f(X_k, X_c)$ are equal expressions, because $X = X_k \cup X_c$.

To estimate the partial dependence, we often use

$$\hat{f}_k(X_k) = \frac{1}{N} \sum_{i=1}^{N} f(X_k, x_{ic}) \,,$$

where $\{x_{1c}, x_{2c}, \ldots, x_{Nc}\}$ are the values of X_c in the training set. This estimation process entails evaluating the function for each value in X_k, which can be computationally demanding. We note that the partial dependence plot is more useful in illustrating the effect of a predictor X_k on the log-odds, when it does not have strong interaction effect with predictors in X_c.

3.4 Performance Metrics

When doing classification, a large number of different metrics are available with which one can judge the performance of a prediction model. It is often useful to consider a confusion matrix in order to define the performance metrics. The confusion matrix in Table 3.1 shows that there are two types of possible errors in classification. A prediction is a *False Negative* (FN), if it has been predicted to be false, but it is in fact true. And similarly, a prediction is a *False Positive* (FP), if it has been predicted to be true but it is in fact false. The relative severity of the two different types of errors is commonly thought to be proportional to the cost of making each of the respective error types.

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

Table 3.1: The confusion matrix

The precision P of a prediction model is defined to be the number of *True Positive* (TP) predictions divided by the number of predictions of type *True*, i.e.

$$P = \frac{TP}{TP + FP} \,.$$

The sensitivity is defined as the proportion of correctly identified true cases, i.e.

$$Sensitivity = \frac{TP}{TP + FN} .$$

And similarly, the specificity is defined as the proportion of correctly identified false cases, i.e.

$$Specificity = \frac{TN}{TN + FP} .$$

There is a trade-off between the specificity and the sensitivity. If one aims for high sensitivity, it usually comes at the cost of lower specificity and vice-versa. The Receiver Operating Characteristics (ROC) curve illustrates this. It plots the sensitivity against the specificity for different discriminatory thresholds. An example of such a curve can be seen in Figure 3.3.

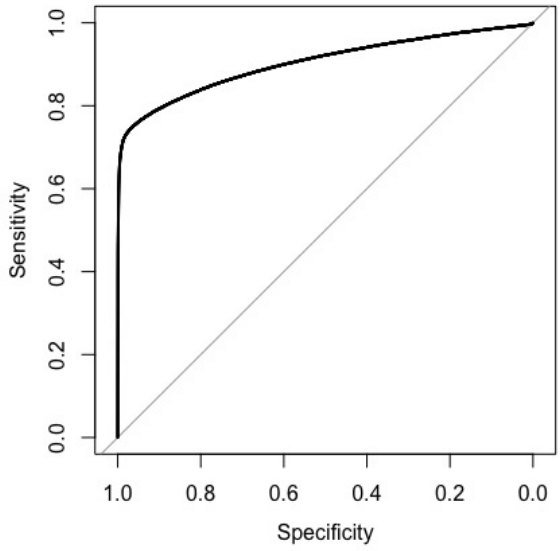


Figure 3.3: Example of a ROC curve.

An idealized ROC curve intersects the point (1,1) in the upper left corner. The prediction model that produces such a curve, makes no prediction errors and high sensitivity does not come at the cost of high specificity. The diagonal line in the ROC curve in the figure represents a prediction model that is equally good as randomly guessing, so an ROC curve which is close to or under the diagonal line represents a poor prediction model. For ROC curves, the *Area Under Curve* (AUC) is another useful metric. It is, as the name suggests,

the area under the ROC curve. For the idealized ROC curve, the AUC is 1. For an ROC curve along the diagonal, the AUC is 0.5. Therefore, the typical AUC value is $(0.5, 1)$, and a good AUC value is close to 1. AUC as a performance metric has the added advantage that it does not require one to choose a discrimination threshold in order to evaluate the performance of a classification model, and it is found to work well as a single number evaluation metric for classification performance [Bradley (1997)].

3.4.1 Lift

As mentioned in the introduction, a common practice in direct marketing is to rank individuals, from most likely to respond, to least likely to respond. To produce a ranking, a classification model must be able to assign probability estimates to the individuals, which in turn can be sorted in a decreasing manner, from highest estimated probability of response, to lowest estimated probability of response. Lift, in the context of direct marketing, can serve as a measure of how good this ranking is.

Assuming a model is able to produce a ranking of a list of prospective customers, the pth-percentile lift of that ranking is defined to be the percentage of respondents in the top pth percentile of the ranking, e.g. if 25% of the respondents are in the top 10th percentile of the ranked list, then the top 10% lift is equal to 25%. Similarly, we can find the top 20% lift by looking at the percentage of respondents in the top 20th percentile of the ranked list, and so on. A model which produces a ranking at random will on average have top 10% lift equal to 10%, and top 20% lift equal to 20%.

In order to evaluate the whole ranking, as opposed to a top percentile of the ranking, Ling and Li (1998) proposed a *lift index* which partitions the ranking into 10 quantiles of equal size and evaluates the cumulative lift for each of them. Let the cumulative lift of the ten quantiles be denoted by $S_1, S_2, \ldots S_{10}$, where S_1 denotes the top 10% lift and S_2 denotes the top 20% lift, and so on. Then the lift index, denoted by S_{lift}, is defined as

$$S_{\text{lift}} = \frac{1 \times S_1 + 0.9 \times S_2 + \ldots 0.1 \times S_{10}}{\sum_i^{10} S_i} .$$

The proposed lift index partitions the ranked examples into 10 quantiles, but for finer partitions, the lift index converges to the area under cumulative lift curve. Figure 3.4 displays a representative cumulative lift curve. The diagonal line in the figure represents randomly guessing, that is, the cumulative lift curve of a model that randomly guesses, will converge to this diagonal line.

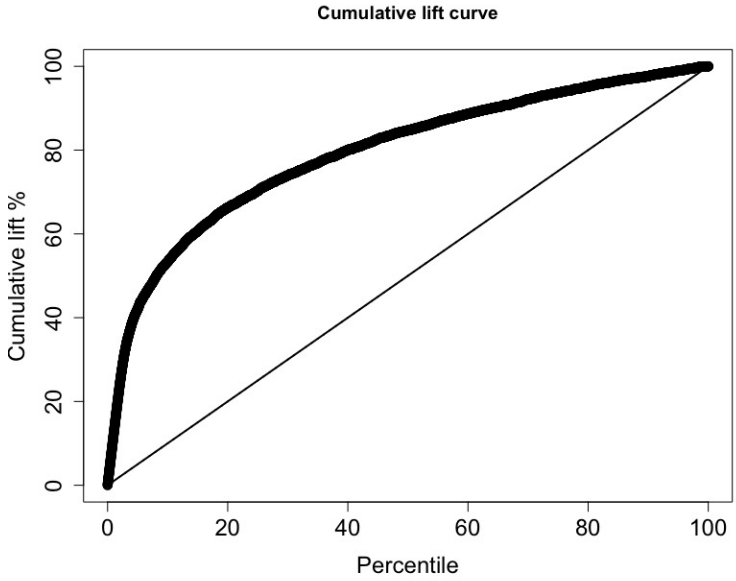


Figure 3.4: Example of a cumulative lift curve.

3.5 Multivariate Control Charts

When a subset of predictors have a particularly large impact on the response, then these predictors could warrant further examination. In particular, it could be interesting to see how they develop over time and to identify trends or patterns. These trends or patterns can then be used to assess whether the distribution of variables has changed or is in the process of changing. Moreover, they can also be used to assess whether it is sensible to retrain the model, for example to use only the most recent data to train the model.

Multivariate quality control charts is one method that can be used to evaluate the stability of a process and to determine if there are any special causes of variation [Johnson et al. (2002)]. In order to do this, we must take into account the correlation between variables, so that we can accurately signal when there is a special cause of variation. The T^2-chart and the *ellipse format* chart are the two most common multivariate control charts.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ be a normally distributed multivariate random variable with mean μ and covariance Σ, then

$$(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) \sim \chi_p^2 \,.$$

is said to be chi-squared distributed with p degrees of freedom.

Let $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be the observed multivariate values and let $\mathbf{S}$ be the associated sample covariance matrix. Furthermore, let

$$\overline{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \, ,$$

then the T^2-statistic for the ith point is defined as

$$T_i^2 = (\mathbf{x}_i - \overline{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x}_i - \overline{\mathbf{x}}) \, .$$

Although $(\mathbf{x}_i - \overline{\mathbf{x}})$ is not independent of $\mathbf{S}$, we can approximate the T^2-statistics to have a χ_p^2 distribution in order to set control limits. The upper control limit can be set to for example $\chi_p^2(0.05)$, which denotes the upper 5% percentile of the χ_p^2 distribution. The points beyond the upper control limit signal that there is a special cause of variation that could warrant attention.

If there is a point that is out of control, i.e. beyond the upper limit, then it is difficult to determine from the T^2-chart alone what variables are responsible. If, however, the multivariate observations only consist of two variables, then ellipse format charts can be helpful to this end.

A 95% quality ellipse consists of all $\mathbf{x}$ that satisfy the inequality

$$(\mathbf{x} - \overline{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x} - \overline{\mathbf{x}}) \leq \chi_2^2(0.05) \, .$$

The ellipse format chart has the two variables along the axes. If there are any points outside the ellipse, it is possible to detect which of the two variables that deviate the most from the average.

Chapter 4

Experiments and Analysis

In this chapter, the models discussed in the previous chapter are implemented. Prior to the implementation, some data processing is performed and a decision is made as to what the training set and test set should be. There is also an error type analysis and a brief discussion on the choice of performance metrics.

4.1 Initial Data Processing

The variable *PNRSerial* denotes the last two digits on the citizenship number. A *PNRSerial* number that is 29 or lower, suggests that the credit card user has recently been granted citizenship. Therefore, a flagging predictor *PNRSerial2* is introduced. The *PNRSerial2* variable is 0 for credit card users who have recently been granted citizenship and 1 for the rest. The original variable *PNRSerial* is removed because it is not assumed to explain the variability beyond what the newly introduced variable *PNRSerial2* can.

There is a column with account numbers in the data set. The account numbers that occur multiple times, represent customers who have been the target of the campaign multiple times. A new flagging variable *OfferedBeforeFlag* is introduced with the motivation that a customer's willingness to respond to the campaign might be affected by him/her being a target for the same campaign earlier, e.g. a customer might be less willing to respond if he or she has already declined the same offer earlier.

A very small proportion of the observations contain missing values. Omitting these observations, we are left with 625179 observations, down from 627113.

4.2 Training Set and Test Set

There is often a decision to be made as to how much data one wants to include in the training and validation process, especially with data going back years in time. Older data may not be as relevant as newer data, and therefore training on older data may reduce the model's predictive performance on future, unseen data. On the other hand, omitting older data may cause the model to miss out on important information. Based on Figure 2.4, displaying how the response rates varies with time, it seems that the rate of response to the marketing campaign has been declining. The response rate for 2018 and 2019 was 0.069, while the response rate for 2015 was 0.187, suggesting that the data from 2015 might not be suitable for training a model to predict on future data.

In order to have a sizable data set for training and validation, without including too much old data, which is believed to be less suitable for prediction, we have chosen to use the data going back 3 years for this thesis, i.e, the data included in the training and validation process is collected between August 2016 and January 2019. The justification for this choice, is that the average response rate for this data is 0.083, which is more akin to the rates observed in the most recent data. Moreover, the data still contains a large number of observations (227910).

This more recent data has been split randomly into a training set consisting of 67% of the data and a test set consisting of the remaining 33% of the data.

4.3 Error Type Analysis

In general, the cost of a false negative is not equal to the cost of a false positive. Therefore, a discussion on the relative cost of different types of errors is often warranted. A false positive represents a customer who was predicted to be a respondent, but who was in fact a non-respondent. Therefore, the cost of a false positive can be said to be the equal to the cost imposing a small inconvenience on a customer.

The other type of prediction error, the false negative, represents a customer who would have responded to the campaign, but was not the target of the campaign. The cost of this error can be said to equal the expected loss of profit obtained, in the case that the customer had applied for a credit limit increase. This expected loss is difficult to quantify, especially because there is a risk involved in increasing the credit limit from the banks point of view.

How these two costs compare is difficult to say. Ultimately it is up to the bank to make a judgement as to how they are going to weigh the different costs. This is usually done with a cost-benefit analysis, but that is beyond the scope of this thesis.

4.4 Choice of Performance Metrics

When doing classification, a large number of different metrics are available with which one can evaluate the performance of a prediction model. As mentioned before, it is a standard practice in direct marketing to rank the possible recipients according to their estimated probability of response. It is not always required that one assigns customers with a probability of response, simply ranking their likelihood of response is often sufficient [Berry and Linoff (2004)]. Following such a ranking, a p-th top percentile of the ranked list is chosen to receive the proposition. This percentile is often chosen according to some profitability analysis.

We recall that the lift is a measure of the model's ability to identify respondents. Choosing 10% lift as the sole performance metric would be fine, assuming that the company only intends to target 10% of their customers, and thus only cares about the number of respondents in the top 10th percentile, but it wouldn't be sufficiently flexible in case the company would want to target a different top percentile of customers. The lift index proposed by Ling and Li (1998) and described in the theory chapter, is flexible in the sense that it is a measure of the lift of the model, but it does not restrict its performance measurement to a predetermined percentile of possible customers, but rather serves as a measurement of how well all the customers are ranked, from most likely to least likely to respond.

The AUC of the ROC curve is another possible candidate to serve as the performance metric, because it does not require one to produce a confusion matrix in order to evaluate the performance, since it works with probability estimates. Coussement et al. (2015) used the AUC as the metric for evaluation to benchmark the predictive performance of common classification techniques on four direct marketing data sets. Miguéis et al. (2017) used the AUC, and the 10% top lift and 20% top lift as the evaluation criteria when modelling the response to direct marketing. But Ling and Li (1998) argue that the cumulative lift curve and the lift index is more intuitive than the AUC in the context of direct marketing. They argue that a lift index serves as a better means of evaluating prediction models.

Since the lift index serves as a measurement of how well the model ranks customers, and is particularly interpretable and intuitive in the context of direct marketing, it is chosen to be the primary performance metric for evaluating models. But since the AUC is a familiar evaluation metric, it will be evaluated as well.

4.5 Fitting the Logit Model

Fitting all covariates using the *glm* package in R, yields error warnings due to linear dependence. The variable *Segment23Name* is the cause of this linear dependence, and omitting it resolves the issue. The resulting model has 71 covariates and we will refer to it as the

full model.

Given that the full model has a large number of covariates, a reduction seems sensible. As discussed in the theory chapter, variable selection can be done with stepwise methods such as by backwards selection with BIC as the selection criterion. Fitting the whole model, we can remove the variable, one-by-one, that results in the greatest reduction in the BIC value.

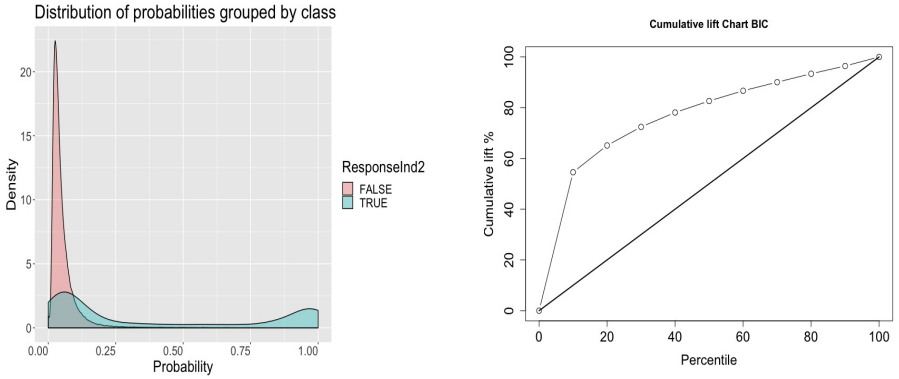
Backward selection is performed and yields a model with 37 parameters including the intercept. We will refer to this as the BIC model. The parameter estimates along with the standard errors, Z-values and the p-values from the Z-tests can be seen in Table 4.1. The R summary of the BIC model can be seen in the appendix.

The only covariate containing categories where the p-value is not smaller than 0.05 is *SCORE*, for which 3 categories have a p-value over 0.05. All the other covariates have a p-value that is smaller than 0.05. As noted in the theory chapter, the way to interpret the regression parameter estimates is through the odds. We recall that the odds for the ith observation are $\frac{P(y_i=1)}{P(y_i=0)}$. We also recall that the odds are scaled by a factor of $\exp \beta_j$ when the value of the jth covariate increases by 1. For the covariate *MonthsSinceAccountCreated*, denoting how long the customer has been with the bank, the parameter estimate is -0.0079. When the value of this covariate increases by 1, the odds are scaled by a factor of $\exp -0.0079 = 0.9921$, i.e. the odds decrease with increasing values of *MonthsSinceAccountCreated*. Since dummy variable coding has been used in fitting this model, the effect of categorical covariates can be evaluated by comparison to the reference category. For example, the covariate *GENDERNAME*, denoting the gender of the customer, has the categories man and $woman$, where $woman$ serves as the reference category. The estimated parameter of Gender:man is 0.2859. The change in odds when gender is changed from $woman$ to man, with all other covariate values unchanged, is equal to a scaling of the odds by a factor of $\exp 0.2859 = 1.3310$. Therefore, the estimated probability of an individual responding to the campaign increases when gender is changed from $woman$ to man, suggesting that men are more likely to respond.

Table 4.1: Parameter estimates for the BIC model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.18	0.126	-17.3	3.15e-67
MonthsSinceAccountCreated	-0.00787	0.000796	-9.89	4.68e-23
GENDER_NAMEMann	0.286	0.0235	12.2	4.19e-34
CreditLimitAmt	0.000265	3.01e-06	88.1	0
HAS_ESTATEMENT_AGREE	0.154	0.0247	6.24	4.45e-10
DaysFirstUse	0.000127	3.31e-05	3.84	0.000123
INTEREST_EARNING_	-7.92e-06	1.85e-06	-4.27	1.93e-05
PrevMinimunToPayAmt	0.000103	2.39e-05	4.32	1.55e-05
BALANCE_AMT	-2.77e-05	1.63e-06	-17	6.96e-65
CASH_BALANCE_AMT	-7.23e-06	1.35e-06	-5.34	9.04e-08
SumTRAVEL_AGENCIESL3	1.22e-05	2.96e-06	4.14	3.5e-05
SumQuasiCashL3	1.26e-05	1.99e-06	6.33	2.41e-10
SumINTERIOR_FURNISHINGS	-9.32e-06	2.01e-06	-4.65	3.31e-06
SumTRAVEL_AGENCIESL12	-7.48e-06	1.88e-06	-3.97	7.05e-05
SUM_of_FirstDunningFlag	-0.083	0.0205	-4.04	5.33e-05
SUM_of_RevolvingFlag	-0.0456	0.00616	-7.4	1.34e-13
SUM_of_FullpayerFlag	-0.0238	0.00629	-3.78	0.000156
SUM_of_OverdraftFlag	0.122	0.0259	4.7	2.54e-06
SUM_of_CreditLimitIncreaseFlag	-0.805	0.0339	-23.8	8.95e-125
SUM_of_CreditLimitDecreaseFlag	2.12	0.0839	25.2	1.74e-140
average_credit_limit_last12	-0.000275	3.17e-06	-86.7	0
avg_rev_bal_L3M	-9.49e-06	1.81e-06	-5.24	1.62e-07
SCORE1	-0.0691	0.103	-0.67	0.503
SCORE2	0.0629	0.104	0.605	0.545
SCORE3	0.209	0.108	1.94	0.0528
SCORE4	0.332	0.111	2.99	0.00278
SCORE5	0.503	0.114	4.39	1.11e-05
SCORE6	0.746	0.128	5.85	5.04e-09
SCORE7	0.74	0.177	4.17	3.07e-05
Distinct_industries	0.0094	0.0026	3.62	0.000297
revUtilizationL12	0.709	0.0884	8.02	1.05e-15
PNRSerial2TRUE	-0.309	0.028	-11	2.98e-28
OfferedBeforeFlagTRUE	-0.2	0.0316	-6.32	2.6e-10
transfercut2. >50	-0.411	0.0543	-7.56	4.08e-14
transfercut3. Never	-0.347	0.0519	-6.68	2.33e-11
purchasecut2. >50	-0.217	0.0417	-5.2	1.96e-07
purchasecut3. Never	-0.0134	0.048	-0.279	0.78

The AUC and lift index can be evaluated for the BIC model. By use of cross-validation, more robust AUC and lift index estimates can be produced. 5-fold cross validation results in an average AUC of 0.8141 and an average lift index of 0.8265. Figure 4.1a shows the distribution of estimated probabilities grouped by the response variable *RespondInd2* on the test set. For non-respondents (red), the estimated probabilities are clearly centered close to 0, which suggests that most non-respondents are assigned a small probability of response. For respondents (blue), there are two peaks, one close to 1 and one close to 0, where the latter is slightly larger.



(a) Distribution of estimated probabilities of the BIC model on the test set.

(b) Cumulative lift of the BIC model on the test set.

Figure 4.1: Test set performance for the BIC model.

Figure 4.1b shows the cumulative lift curve using 10 partitions of the test set. The model is able to capture more than 50% of the respondents in the top 10% of ranked customers, and around 65% of the respondents in the top 20% of ranked customers. On the test set the BIC model recorded a lift index of 0.8193.

4.6 Regularizing the Logit Model

We recall that shrinkage methods can be used to reduce the variance at the cost of a small increase in the bias. A common way to apply shrinkage, is to add a penalty term to the objective function that penalizes the size of the parameter estimates. The form of the penalty term affects how the parameters are constrained. The L_1 penalty was introduced in chapter 3. It has the property that it performs variable selection, in addition to shrinking the coefficients. The size of the penalty hyperparameter λ determines how many predictors are included in the model. One possible strategy to select an appropriate value for λ, is to perform cross-validation on a grid of values for λ, and select the value of lambda for

which the average lift index is highest. Following this strategy, models were trained using the *train* function in the *caret* package with the *glmnet* implementation [Kuhn (2019)].

The result of using this training procedure with 5-fold cross validation can be seen in Figure 4.2. The dotted red line indicates the value of λ that achieved the highest lift index. The grey area indicates the estimated standard deviation. This implementation uses the cyclical coordinate descent procedure to find the regularized parameter estimates.

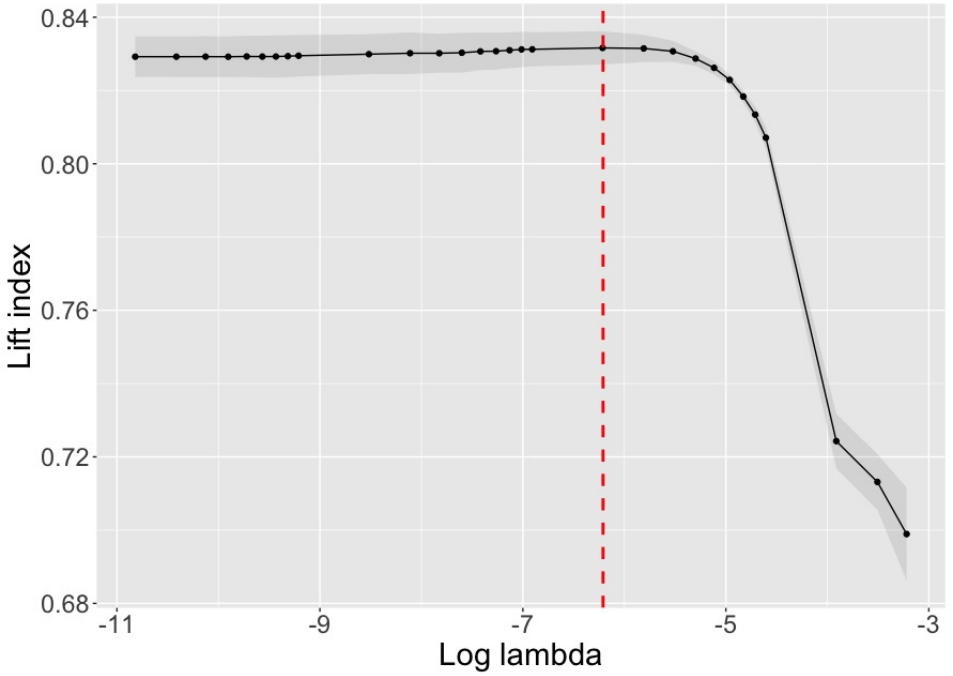


Figure 4.2: Cross-validation results on different values of λ.

The value of λ that achieved the highest lift index was $\lambda = 0.002 = \exp(-6.21)$, for which the lift index was 0.8315, but any $\log \lambda \in [-6, -11]$ has an associated lift index which is close to 0.8315. It can be argued that among a set of equally-performing λ-values, that the largest value for λ ought to be chosen. The number of predictors in the model is smaller for larger values of λ, and the smaller model is preferred over the larger model when their predictive performance is equal. Therefore, we will proceed with the model with $\lambda = 0.002 = \exp(-6.21)$ in this thesis.

The variable selection property and the coefficient shrinking property of the L_1 penalty term can be illustrated by plotting the coefficient values against the λ value. This plot can be seen in Figure 4.3. The figure illustrates that the coefficient values go to 0 as λ increases. The optimal λ value is marked by the dotted red line. With this value for λ, 39 covariates

are included in the model. The coefficient values are stated in the appendix.

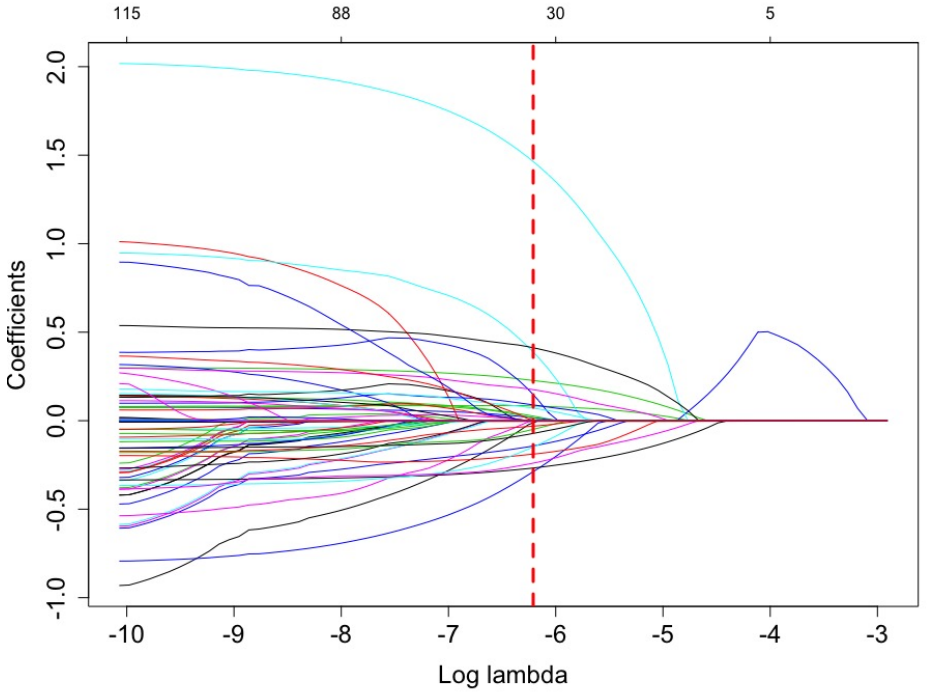
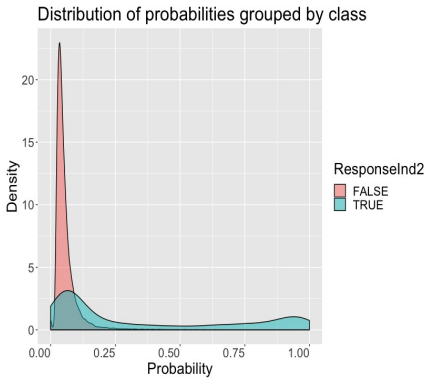


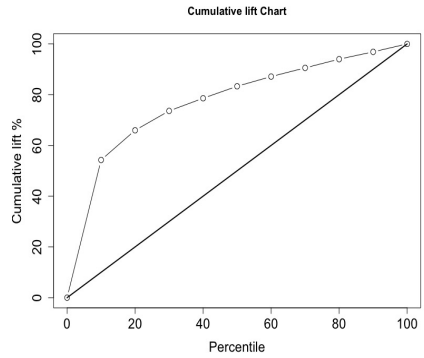
Figure 4.3: Regression coefficients for different values of λ.

Figure 4.4a shows the distribution of estimated probabilities grouped by class. This distribution is similar to the distribution obtained with the BIC model, although the probabilities for the respondents seem to be a little more spread out, an effect which is often observed when shrinkage is applied to the parameters.

Figure 4.4b shows cumulative lift on the test set with 10 partitions. The figure shows that the model is able to capture more than 50% of the respondents by targeting the top 10% of ranked customers. To illustrate what this means, we consider the case when the company does indeed choose to target the top 10% of ranked customers. That corresponds to targeting 7521 customers, of which 3377 are respondents and 4144 are non-respondents, which corresponds to a precision of $P = \frac{3377}{7521} = 0.449$. The lift index, which is closely related to the area under the cumulative lift curve, was 0.8242 on the test set.



(a) Distribution of estimated probabilities for the regularized logit model.



(b) The cumulative lift for the regularized logit model.

Figure 4.4: Test set performance for the regularized logit model.

4.7 Random Forests

We recall that random forests are a modified version of bootstrap aggregated decision trees, where only a random subset of predictors are available per split.

Random forest models were trained using the *train* function in the *caret* package with implementation *Ranger* in R, which is a fast implementation of random forest [Wright and Ziegler (2017)]. The results of using a tuning grid and 5-fold cross validation can be seen in Table 4.2. The estimated standard deviations for the lift index values and the AUC are also stated in the table. Note that the ROC column denotes the AUC values.

Table 4.2: Hyperparameter tuning results for random forests

m	min.node.size	ROC	LiftIndex	ROCSD	LiftIndexSD
6	3	0.7928	0.8098	0.0029	0.0027
6	6	0.7925	0.8099	0.0036	0.0033
6	9	0.7927	0.8101	0.0028	0.0024
12	3	0.8175	0.8318	0.0025	0.0024
12	6	0.8167	0.8307	0.0028	0.0030
12	9	0.8170	0.8313	0.0028	0.0025
18	3	0.8223	0.8346	0.0033	0.0032
18	6	0.8225	0.8348	0.0036	0.0039
18	9	0.8225	0.8349	0.0037	0.0031

The values considered for m, denoting the number of predictors sampled at each split,

were m=(6,12,18). The values considered for n_{min}, denoting the minimum number of training examples in the terminal nodes, were n_{min}=(3,6,9). For each of the 9 combinations of hyperparameteres, 500 trees were trained.

Among the hyperparameter combinations tested, the one with $m = 18$ and $n_{min} = 9$ resulted in the highest average liftindex, which was 0.8349. The average AUC value with this particular combination was 0.8225. The lift index was consistently lower for $m = 6$, than for the higher values for m. When the number of variables is large, but the number of relevant variables is small, random forests tend to perform poorly for low values of m [Friedman et al. (2001)]. The fact that larger values of m performed better in cross-validation suggests that the number of relevant variables is small in comparison to the total number of variables.

The hyperparameter tuning results are visualized in Figure 4.5. Judging by the figure, it seems that the different values for the number of selected predictors m, has a larger influence on the lift index than the minimal node size.

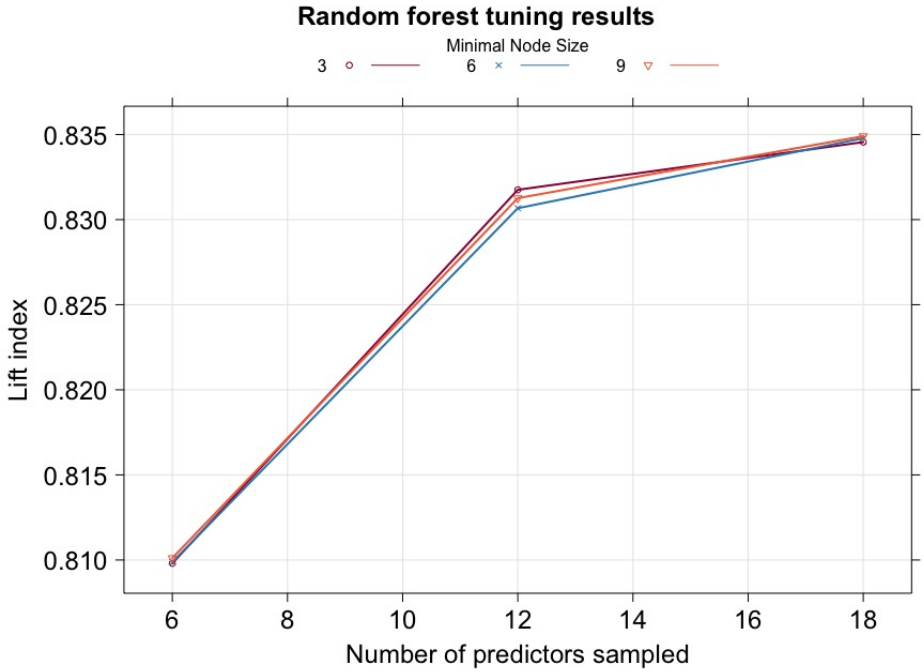


Figure 4.5: The random forest hyperparameter tuning results.

In order to assess how much predictors affect the response, we can look at the relative importance of the predictors. We find the importance of a predictor by looking at how much node purity is increased by the predictor's splits over all the trees. The relative

importance of predictors can be seen in Figure 4.6. The predictors' importance is scaled to the credit limit amount, which was the most important predictor. The second most important predictor, the average credit limit over the 12 months prior to the campaign, is highly correlated with the credit limit amount. The third most important predictor was the balance amount. Some other notable inclusions in the top 20 most important predictors are the number of days before the credit card is used, the customers age and the number of months since the account was created.

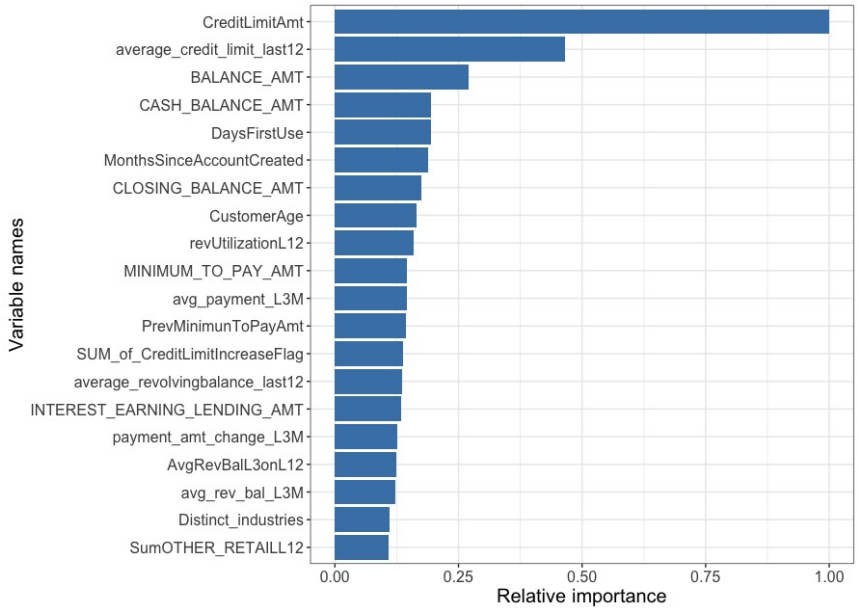


Figure 4.6: The relative importance of variables for the random forest model.

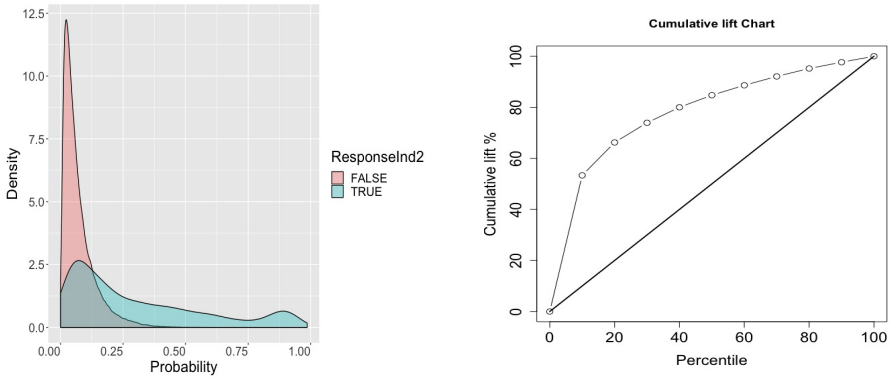
By this plot alone, it is difficult to tell how the variables affect the response, but we saw in Figure 2.2 that the response rate was higher for individuals with already high credit limit amounts. This suggests that individuals with high credit limits will be estimated to have a high probability of response, according to the random forest model.

The distribution of estimated probabilities on the test set, grouped by the response, can be seen in Figure 4.7a. The non-respondents are centered close to 0, and the respondents are more spread out, but still skewed a little towards 0. The respondents being more spread out in the random forest model than the respondents in the logit models, is likely to be a result of the way that random forests assign probability estimates. Individually, the trees assign a classification of either 0 or 1, then the probability estimate is obtained by dividing the number of 1 classifications with the total number of classifications. The random forest model differs in this respect with the logit model and the GBM model, both of which

produce log-odds estimates.

The cumulative lift chart can be seen in Figure 4.7b. The figure shows that if one were to use the random forest model to rank the individuals in the test set, and target for example the top 50% of the individuals, then around 80% of the respondents would be in that target group.

The recorded lift index value on the test set for the random forest model was 0.8318.



(a) Random forest probability distribution on test set

(b) Random forest cumulative lift chart on the test set

Figure 4.7: Test set performance for the random forest model.

4.8 Gradient Boosting Machine

We recall that training gradient boosting machines (GBM) entail sequentially fitting regression trees to the residuals of the preceding trees, and updating the residuals according to a learning rate. The gradient boosting models were trained using the *train* function in the *caret* package with the *gbm* implementation [Greenwell et al. (2019)].

As discussed earlier, a possible approach to hyperparameter tuning was to select a large, albeit computationally feasible, number of trees and try different small learning rates, to shrink the contribution of each tree. This approach takes into account the particular relation between the number of trees and the learning rate that was discussed in the theory chapter. Employing this approach, the number of trees was set to 2500, and the other hyperparameters were allowed to vary.

Three hyperparameters were considered in tuning, namely the learning rate λ, the interaction depth d and the minimum training observations per node n_{min}. The learning rate values considered were $(0.05, 0.1, 0.2)$, the interaction depth values considered were $(1, 2, 3)$ and the values considered for the minimum number of observations per node were $(5, 10, 20)$. All possible combinations of these hyperparameters were used, which resulted

in $3^3 = 27$ different models. The number of trees was 2500 for all 27 models.

The results of using a tuning grid with 5-fold cross validation can be seen in Table 4.3. We note that the the learning rate is often referred to as the *shrinkage* parameter, due to the fact that it shrinks the contribution from each tree. Furthermore, the average AUC values are stated in the column called *ROC* and the two last columns in the table state the estimated standard deviation for both the AUC and lift index values.

Table 4.3: Hyperparameter tuning results for GBM.

shrinkage	depth	n.min	n.trees	ROC	LiftIndex	ROCSD	LiftIndexSD
0.05	1	5	2500	0.8238	0.8362	0.0060	0.0054
0.05	1	10	2500	0.8242	0.8363	0.0061	0.0053
0.05	1	20	2500	0.8242	0.8362	0.0064	0.0058
0.10	1	5	2500	0.8265	0.8376	0.0057	0.0049
0.10	1	10	2500	0.8261	0.8372	0.0053	0.0048
0.10	1	20	2500	0.8262	0.8372	0.0057	0.0049
0.20	1	5	2500	0.8253	0.8364	0.0052	0.0047
0.20	1	10	2500	0.8249	0.8362	0.0058	0.0051
0.20	1	20	2500	0.8243	0.8355	0.0047	0.0041
0.05	2	5	2500	0.8334	0.8431	0.0050	0.0045
0.05	2	10	2500	0.8331	0.8431	0.0047	0.0043
0.05	2	20	2500	0.8333	0.8433	0.0050	0.0041
0.10	2	5	2500	0.8307	0.8408	0.0041	0.0038
0.10	2	10	2500	0.8303	0.8406	0.0040	0.0036
0.10	2	20	2500	0.8301	0.8403	0.0045	0.0038
0.20	2	5	2500	0.8225	0.8336	0.0031	0.0027
0.20	2	10	2500	0.8221	0.8331	0.0027	0.0024
0.20	2	20	2500	0.8220	0.8330	0.0029	0.0025
0.05	3	5	2500	0.8361	0.8454	0.0060	0.0053
0.05	3	10	2500	0.8358	0.8455	0.0052	0.0050
0.05	3	20	2500	0.8361	0.8458	0.0051	0.0048
0.10	3	5	2500	0.8316	0.8417	0.0048	0.0046
0.10	3	10	2500	0.8318	0.8419	0.0040	0.0040
0.10	3	20	2500	0.8302	0.8403	0.0054	0.0049
0.20	3	5	2500	0.8173	0.8288	0.0049	0.0045
0.20	3	10	2500	0.8154	0.8272	0.0031	0.0030
0.20	3	20	2500	0.8166	0.8276	0.0013	0.0011

Figure 4.8 visualizes the hyperparameter tuning results. Looking at the figure we note that for an interaction depth of 1, the models with larger learning rates performed slightly

better. Allowing for higher order interaction effects with $d > 1$, the models with $\lambda = 0.05$ performed consistently better than the models with higher learning rates. As with the random forest hyperparameter tuning, the minimum observations per node did not seem to have a large impact on the performance. One can see on the figure of the tuning results that the lift index does not vary much across the three different values $(5, 10, 20)$ for the minimum node size. The best result was obtained for $(\lambda, d, n) = (0.05, 3, 20)$, for which the average lift index was 0.8458 and the average AUC was 0.8361 in cross-validation. This is the model we will proceed with. The model learns slowly with a learning rate of only 0.05, compared to the default which is usually 0.1. We also note that $d = 3$ means that the model allows for third order interaction effects.

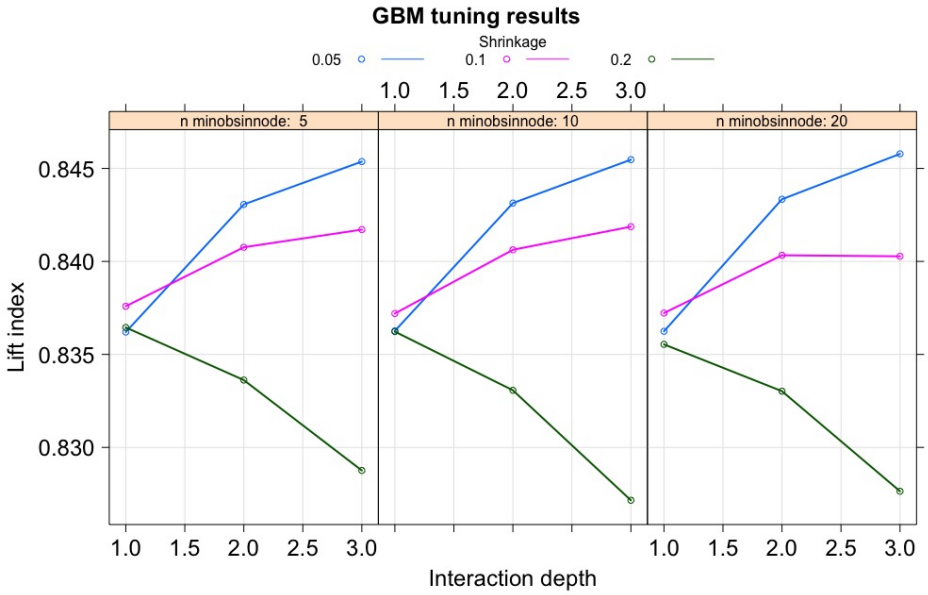
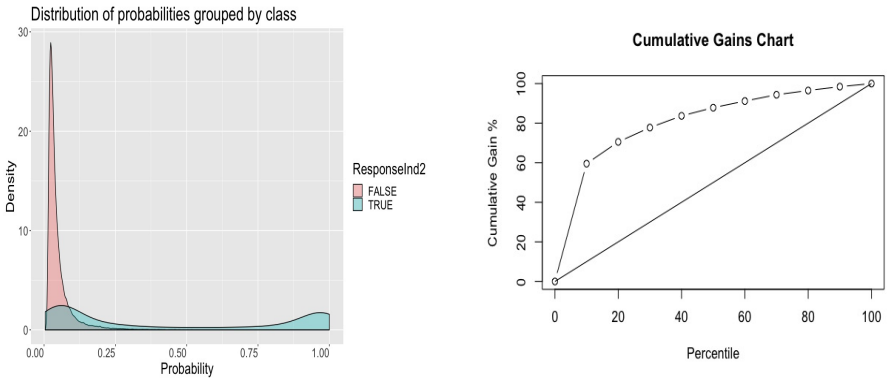


Figure 4.8: The hyperparameter tuning results for GBM.

The distribution of estimated probabilities on the test set grouped by the response can be seen in Figure 4.9a. The non-respondents are centered around 0, and the respondents are distributed with two small peaks around 0 and 1. This distribution is similar to the distribution we have seen for the logit models. As with the logit models, the GBM model estimates the probability of a large proportion of non-respondents to be close to 0, but there is also a large proportion of respondents whose probability is estimated to be close to 0.

The cumulative lift chart can be seen in Figure 4.9b. The plot shows that the model is able to capture around 60% of respondents in the 10% highest ranked customers.

The recorded lift index for the GBM model was 0.8598 on the test set.



(a) GBM probability distribution on the test set. **(b)** GBM cumulative lift chart on the test set.

Figure 4.9: Test set performance for the GBM model.

We can produce an overview of the relative importance of the predictors in the GBM model, by looking at how much each predictor contributes to reducing the squared errors in the B regression trees. The relative importance of the 20 top predictors for GBM model can be seen in Figure 4.10. As with random forests, the credit limit is the most important predictor. The second most important variable, the average credit limit over the last 12 months, is highly correlated with the credit limit. In fact, their correlation is 0.948. The top 20 predictors for gradient boosting resemble the top 20 predictors for random forest. Some notable deviations include the *SUM_of_CreditLimitIncreaseFlag* variable, which denotes the number of times the customer has received a credit limit increase. This variable is is placed 3rd for the gradient boosting model, as opposed to 13th for the random forest model.

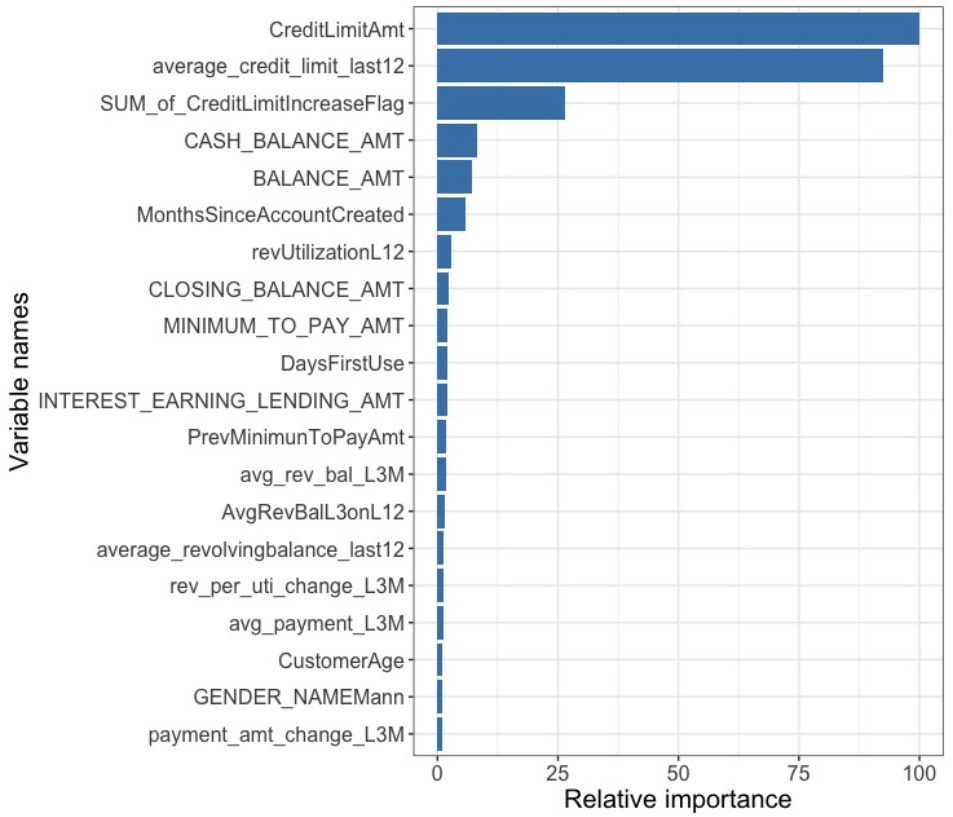
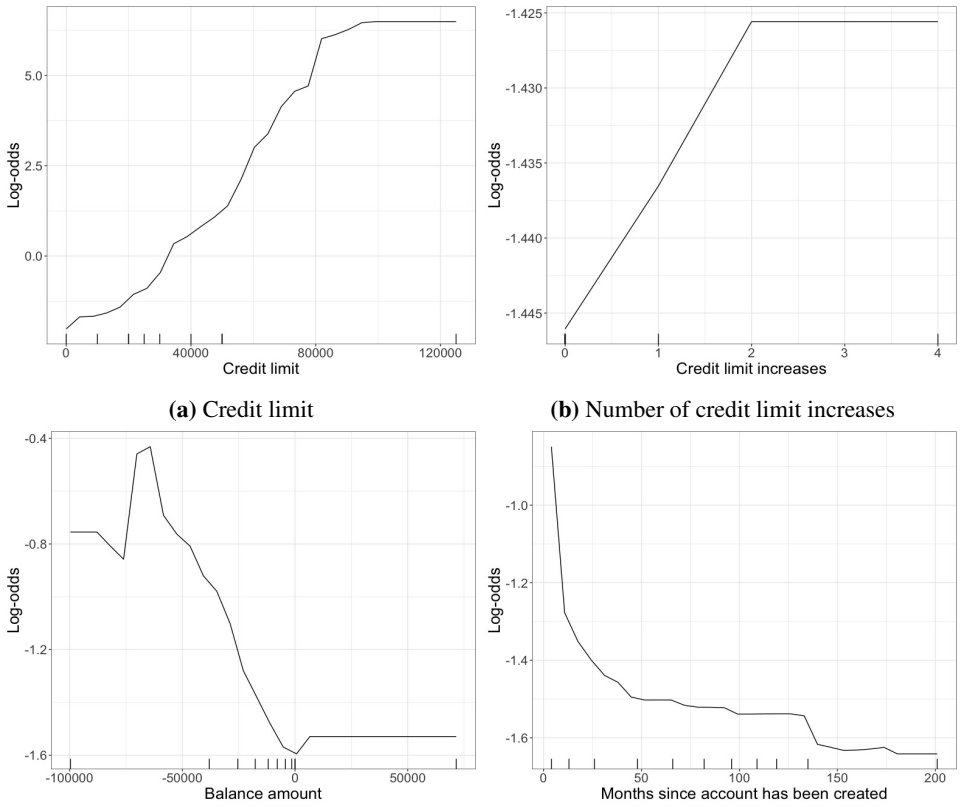


Figure 4.10: The relative importance of variables for the GBM model.

The relative importance allows us to consider what predictors warrant further analysis. Partial dependence plots allow us to analyze the effect of these predictors on the log-odds. The plots are produced by estimating the marginal average of the predictor on the response. Figure 4.11 shows the partial dependency plots for some of the most important predictors, produced by the *pdp* package in *R* [Greenwell (2017)].



(a) Credit limit

(b) Number of credit limit increases

(c) Balance

(d) Months since account created

Figure 4.11: The partial dependency plots for individual predictors.

The ticks in the bottom of the plots represent the deciles of the distribution of training examples. The intervals where the deciles are dense warrant special attention, because this is where the majority of the data is.

The partial dependence plot for the credit limit can be seen in Figure 4.11a. The plot shows how the log-odds vary with different credit limits, and from the plot it is evident that the log-odds increase rapidly with increasing credit limit amount. This is in line with the understanding that the credit limit is the most important predictor, and in line with the understanding that the likelihood of response increases with increasing credit limit amount.

Figure 4.11b shows how the log-odds vary with different number of credit increases prior to the campaign. The log-odds increase when the *SUM_of_CreditLimitIncreaseFlag* increases from 0 to 2, but does not vary beyond that, suggesting that the likelihood of response increases when the number of prior limit increases goes from 0 to 1 or 1 to 2.

Figure 4.11c shows how the log-odds vary with different values of the balance amount.

The figure shows that the log-odds are lower for larger values of the balance amount, suggesting that the likelihood of response decreases when the value of the balance amount increases.

Figure 4.11d shows the partial dependence of the predictor *MonthsSinceAccountCreated*, denoting the number of months since the account has been created. It shows that the log-odds is higher for smaller number of months, suggesting that the likelihood of response decreases when the number of months increases.

Figure 4.12 shows the partial plot for both the credit limit and the balance amount, two of the most important predictors for both the random forest model and the GBM model. This plot allows us to assess the joint effect of the two predictors. Judging by the plot, there is no immediately visible interaction effect between the two variables. The plot does however also allow us to assess the difference in influence between the two variables. The effect of varying values of the credit limit on the log-odds seems to be much larger than the effect of varying values of the balance amount.

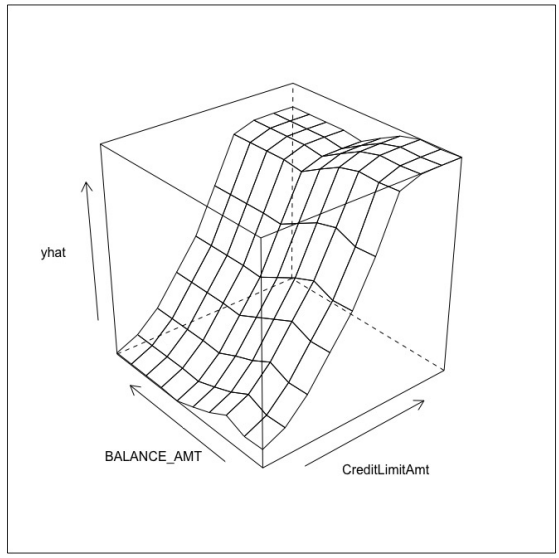


Figure 4.12: Partial plot for credit limit and balance amount.

4.9 Comparing the Models

We've used three different types of models in this thesis, now we would like to compare their lift index performance to see if there is a significant difference. For each of the models, we've used the hyperparameter combinations that obtained the best lift index values. Since the regularized logit model performed better than the BIC model, it is chosen to represent the logit model when comparing model performance.

The basis for comparing the models, is the performance of the models on the left-out validation set in the 5-fold cross-validation experiment. Each of the five folds in cross-validation serve as validation sets exactly once, resulting in five lift index values for each models. The results are stated in Table 4.4.

Table 4.4: 5-fold cross-validation performance on the left-out validation sets.

	GBM	Random forest	Logit
1	0.8500	0.8381	0.8370
2	0.8451	0.8337	0.8376
3	0.8422	0.8418	0.8255
4	0.8404	0.8400	0.8262
5	0.8513	0.8369	0.8313

Based on the numbers in the table, it seems that GBM performed better than the other models in cross-validation. In order to assess whether the difference in performance is significant, we can do paired comparisons of the means of the lift index values. For a paired comparison between means μ_i and μ_j, the null-hypothesis and alternative hypothesis are

$$\begin{aligned} \mathrm{H}_0 &: \mu_i - \mu_j = 0 \\ \mathrm{H}_1 &: \mu_i - \mu_j \neq 0 \end{aligned}$$

respectively. If we reject the null-hypothesis, then we say that the difference in means is significant. If we don't reject the null-hypothesis, then we say that there isn't a significant difference in the means.

In order to test the hypotheses, we are interested in the difference between the models' performance. The differences is stated in Table 4.5.

Table 4.5: Differences in cross-validation.

	GBM diff Random forest	GBM diff logit	Random forest diff logit
1	0.0173	0.0179	0.0006
2	0.0017	0.0037	0.0021
3	0.0123	0.0207	0.0084
4	0.0088	0.0232	0.0145
5	0.0092	0.0174	0.0082

The *t-test* allows us to conduct the paired comparisons. Let $\overline{d}$ denote the average difference between two means, and let s denote the sample standard deviation and let n denote the number of samples.

Then the test-statistic for the t-test is

$$T = \frac{\overline{d}}{\frac{s}{\sqrt{n}}} .$$

Under the null-hypothesis, the test-statistic T follows a t-distribution with $n - 1$ degrees of freedom. We can compare the value of the test-statistic with the t-distribution to obtain a p-value for the test. We then reject the null-hypothesis if the obtained p-value is smaller than a chosen significance level α.

When conducting multiple tests, we ought to take into account the family-wise error rate, i.e. the rate of falsely rejecting one or more null-hypotheses. The *Bonferroni correction* adjusts the p-values, by accounting for the number of paired comparisons, to control the family-wise error rate at α. The observed p-values are divided by the number of tests to obtain Bonferroni-corrected p-values.

In our case, we have 3 tests, so the observed p-values are divided by 3. The Bonferroni-corrected p-values from our 3 t-tests are stated in Table 4.6.

Table 4.6: Bonferroni-corrected p-values obtained from the t-tests.

GBM diff Random forest	GBM diff logit	Random forest diff logit
0.189949	0.006936	0.469454

Using $\alpha = 0.05$ as our significance level, we reject only one null-hypothesis, namely the null-hypothesis that the GBM mean and logit mean are equal. We conclude that the GBM model performs significantly better than the logit model. We cannot reject the other two null-hypotheses, which means that we cannot say that there is a significant difference in performance between the GBM model and the random forest model, and we cannot say that there is a significant difference in performance between the logit model and the random forest model.

4.10 Statistical Process Control

We saw that two of the most important variables according to the random forest model and the GBM model was the credit limit and the balance amount. Given that these variables seem to affect the response a lot, it could be interesting to analyze their development over time. Gama et al. (2004) proposed a method for detecting model drift, i.e. detecting changes in the distribution of the data. Even though multivariate control charts were not mentioned, they can also be used to to detect change in the distributions of key variables.

The T^2-chart is a common control chart used to see how variables vary over time, and can thus be used to identify trends or patterns. We recall from chapter 3 that the T^2-statistic for the ith point is defined as

$$T_i^2 = (\mathbf{x}_i - \overline{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x}_i - \overline{\mathbf{x}})$$

where $\mathbf{S}$ is the sample covariance matrix for the variables involved. The T^2-values can then be plotted on a time axis.

The sample covariance matrix is

$$S = \begin{pmatrix} 2615806 & -1545301 \\ -1545301 & 2632963\,. \end{pmatrix}$$

The covariance between the two variables is negative, which means that there is a negative correlation between the credit limit and the balance amount. We note that the balance amount tends to be negative, because credit cards tend to have a negative balance.

The T^2-chart for the variables credit limit and balance amount for the nine campaign periods featured in the training data, i.e. the nine campaign months stretching from August 2016 to January 2019, can be seen in Figure 4.13.

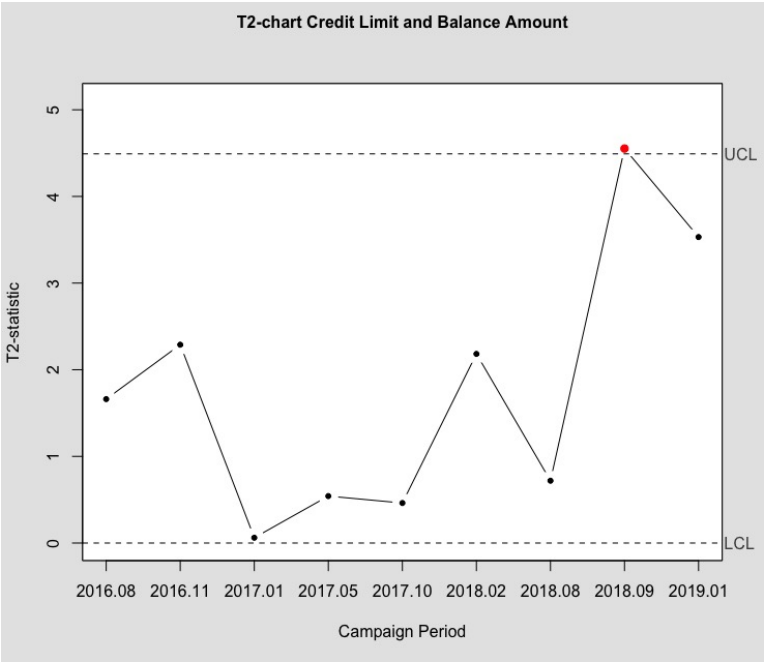


Figure 4.13: T^2 Control chart for credit limit and balance amount with upper control limit (UCL) set to $\chi^2_2(0.05)$.

The quality level for the chart is 0.95. The plot is produced by averaging the values for each of the nine time periods, i.e. for the first time period, which is August 2016, the average value for the credit limit and the balance amount are used to create $\mathbf{x}_1$, and $\mathbf{x}_2, \mathbf{x}_3, \ldots \mathbf{x}_9$ were created using data from the other eight campaign periods.

A single point is found to be larger than the upper control limit. This point corresponds to the second last campaign period, which is September 2018. The T^2-statistic for the last campaign period was the second largest, but still within the upper control limit. The fact that the two largest T^2-statistics occur in the two last campaign periods, could suggest that the distributions for these key variables are changing, i.e. that future data could have a different distribution for the credit limit and balance amount, compared to the distributions that were used to train the models. This could raise concerns for the model's validity in predicting on future data.

The ellipse format chart is another common multivariate control chart. If there is irregular behaviour, it can give insight into the question of which variables deviate from their average. The ellipse chart can be seen in Figure 4.14.

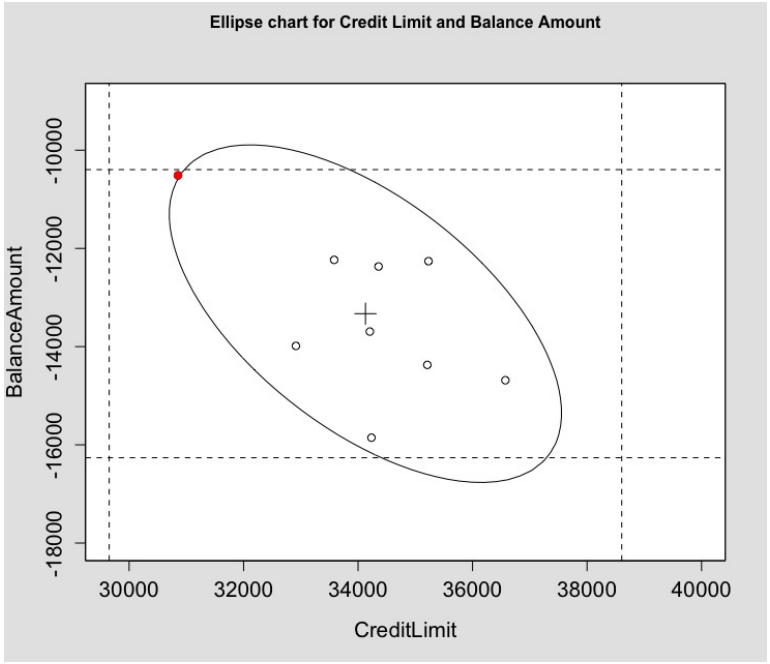


Figure 4.14: A 95% quality ellipse based credit limit and balance amount.

The single out-of-bounds point is marked by the red dot. For this particular point, the observed average credit limit is particularly low, compared to the the others. The credit limit for this point is 31000 kroner, whereas the average for the nine points is 34000 kroner. The balance amount was also found to be larger than the average. The average balance amount is around -13500 kroner, and the single out-of-bounds point has a balance amount of around -10500 kroner.

Chapter 5

Summary, Discussion and Conclusions

The aim of this thesis was to produce models to predict direct marketing response using statistical learning methods. Special emphasis was placed on exploring the benefits of model tuning and model modification, as well as exploring the different models' interpretation tools to better understand the relationship between the explanatory variables and the response. To this end several models were fitted.

The logit models were fitted using both backward selection and L_1 regularization to perform model selection. Cross-validation was employed to select an optimal value for the regularization hyperparameter, and its effect on the coefficients was discussed. Although the logit model is not always on par with some of the common non-parametric models in terms of predictive performance, it is often regarded as being easier to interpret. The effect of individual predictors can be analyzed by looking at their effect on the odds. The lift index for the BIC model was 0.8193 and the lift index for the regularized logit model was 0.8242 on the test set.

Random forest models were fitted using cross-validation to determine optimal values for the number of predictors sampled at each node split and the minimum leaf node size. To better understand the effect of predictors on the response, an overview of the relative importance of predictors was produced. The credit limit amount was by far the most important predictor. Not including highly correlated predictors, the balance amount and the number of days before use, were also ranked high on this list. The lift index for the random forest model was 0.8318 on the test set.

The gradient boosting machine was also used to produce a prediction model. Although superficially similar to random forests, GBMs represent a fundamentally different

approach, due to the fact that trees are fitted *sequentially* to the residuals of the preceding trees. The chosen strategy in fitting the model was to find a computationally feasible fixed number of trees and consider different learning rates. The interaction depth controls the maximum number of splits per tree, and thus also controls the level of the interaction effects. In cross-validation, the best result was obtained with an interaction depth of 3, allowing third-order interaction effects to be modelled.

The gradient boosting machine obtained the best predictive performance among the models fitted. Using the best-performing hyperparameters from cross-validation, the lift index was 0.8598 on the test set. In addition to an overview of the most influential predictors, the partial dependency plots for a selection of predictors was produced, showing the effect the predictors have on the log-odds. A bivariate partial dependency plot was also produced for credit limit and balance amount, in order to assess the interaction effect between two of the most important predictors.

We have compared the predictive performance of the models. The means of the lift index values from cross-validation was compared. Using paired t-tests, GBM was found to perform significantly better than the regularized logit model, while controlling the rate of falsely rejecting on or more null-hypotheses at $\alpha = 5\%$. No significant difference in performance was found between the random forest model and the GBM model. Furthermore, no significant difference in performance was found between the logit model and the random forest model.

Both the random forest model and the GBM model suggested that the credit limit amount and balance amount were highly influential predictors. If the goal is to predict on unseen data, it is argued that studying the behaviour of these two variables for possible trends and special causes of variation is warranted. Multivariate control charts were used to this end. The T^2-chart identified a single time period out of the possible 9 time periods that was outside the 95% quality level. This was the second last time period corresponding to the campaign period of September 2018. The T^2-statistic of the last campaign period, January 2019, also seemed to deviate from the average, but was still within the 95% quality level. The last two T^2-statistics could suggest that future data will differ significantly on important variables, thus raising concerns regarding the validity of the model's predictions on future data. One possible way to deal with this is to use only the most recent data in training. Even though this approach might come at the cost of losing valuable information, the most recent data could be used to train a better-performing model when the aim is to predict on future data.

5.1 Discussion on Future of Credit Cards

Citing increasing consumer debt, *Finanstilsynet*, an independent government agency, suggested several measures to curb the development [Finanstilsynet (2018)]. Among these suggested measures was a debt register to, amongst other things, track information on active credit cards. Such a measure was implemented, and as of July 2019, banks could access more complete information on the debt situation on individuals applying for a loan or a credit limit increase [Regjeringen (2019)]. This is believed to increase the number of declined requests for credit limit increases. Furthermore, it is possible that some individuals are less inclined to apply for a credit limit increase if they believe their request will be denied, thus negatively affecting the response rate.

Traditional credit cards could also face competition from new fintech solutions. The popular mobile payment application *Vipps* is looking to extend credit for its users [Lorentzen, Marius (2019)]. It is possible that some customers will feel that they get their need for credit satisfied through *Vipps*, and therefore will feel less inclined to apply for a credit limit increase, which could adversely affect the response rate for the campaign.

5.2 Recommendations for Further Work

It is possible to consider macroeconomic variables as well. The value of the Norwegian krone, or the interest rate are possible explanatory variables. For example, if the krone is weak or continues to weaken, that could positively affect the response rate.

Although imbalanced data sets are not a major issue when the aim is to *rank* the observations, it could be interesting to explore different re-balancing techniques, to see if improvements can be made to the predictive performance.

One could also try fundamentally different methods to produce a prediction model. If the main focus is predictive performance, rather than interpretation, one could for example explore artificial neural networks or support vector machines as methods to rank the prospective customers.

Bibliography

Baesens, B., Viaene, S., Van den Poel, D., Vanthienen, J., Dedene, G., 2002. Bayesian neural network learning for repeat purchase modelling in direct marketing. European Journal of Operational Research 138 (1), 191–211.

Berry, M. J., Linoff, G. S., 2004. Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons.

Bradley, A. P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern recognition 30 (7), 1145–1159.

Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140.
URL https://doi.org/10.1007/BF00058655

Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

Coussement, K., Harrigan, P., Benoit, D., 2015. Improving direct mail targeting through customer response modeling. Expert Systems with Applications 42 (22), 8403–8412.

Fahrmeir, L., Kneib, T., Lang, S., Marx, B., 2013. Regression: models, methods and applications. Springer Science & Business Media.

Finanstilsynet, 2018. Finanstilsynet foreslår forskriftsregulering av forbrukslån.
www.finanstilsynet.no/nyhetsarkiv/pressemeldinger/2018/finanstilsynet-foreslar-forskriftsregulering-av-forbrukslan/.

Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. Vol. 1. Springer series in statistics New York.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33 (1), 1–22.
URL http://www.jstatsoft.org/v33/i01/

Friedman, J. H., 2001. Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29 (5), 1189–1232.
URL http://www.jstor.org/stable/2699986

Gama, J., Medas, P., Castillo, G., Rodrigues, P., 2004. Learning with drift detection. In: Bazzan, A. L. C., Labidi, S. (Eds.), Advances in Artificial Intelligence – SBIA 2004. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 286–295.

GDPR, 2016. Gdpr.
https://gdpr-info.eu/issues/email-marketing/.

Greenwell, B., Boehmke, B., Cunningham, J., Developers, G., 2019. gbm: Generalized Boosted Regression Models. R package version 2.1.5.
URL https://CRAN.R-project.org/package=gbm

Greenwell, B. M., 2017. pdp: an r package for constructing partial dependence plots. The R Journal 9 (1), 421–436.

Johnson, R. A., Wichern, D. W., et al., 2002. Applied multivariate statistical analysis. Vol. 5. Prentice hall Upper Saddle River, NJ.

Kuhn, M., 2019. caret: Classification and Regression Training. R package version 6.0-84.
URL https://CRAN.R-project.org/package=caret

Ling, C. X., Li, C., 1998. Data mining for direct marketing: Problems and solutions. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. KDD'98. AAAI Press, pp. 73–79.
URL http://dl.acm.org/citation.cfm?id=3000292.3000304

Lorentzen, Marius, 2019. Vil formidle kreditt: Nå vil vipps legge ut for regningene dine.
www.e24.no/boers-og-finans/i/awekBd/vil-formidle-kreditt-naa-vil-vipps-legge-ut-for-regningene-dine/.

Miguéis, V. L., Camanho, A. S., Borges, J., 2017. Predicting direct marketing response in banking: comparison of class imbalance methods. Service Business 11 (4), 831–849.

Natekin, A., Knoll, A., 2013. Gradient boosting machines, a tutorial. Frontiers in neurorobotics 7, 21.

Norske lover, 2009. Markedsføringsloven.
https://lovdata.no/dokument/NL/lov/2009-01-09-2.

Regjeringen, July 2019. Gjeldsinformasjonsloven.
URL https://www.regjeringen.no/no/tema/forbruker/gjeldsinformasjonsloven/id2510537/

Schwarz, G., 03 1978. Estimating the dimension of a model. Ann. Statist. 6 (2), 461–464. URL https://doi.org/10.1214/aos/1176344136

Wright, M. N., Ziegler, A., 2017. ranger: A fast implementation of random forests for high dimensional data in C++ and R. Journal of Statistical Software 77 (1), 1–17.

Appendix

Table 5.1: A description of the variables

Variable Name	Description
BK_ACCOUNT_ID	Internal account number
YearMonth	Year and month in format YYYYMM
CustomerAge	Customer's age in years
MonthsSinceAccountCreated	Account's age in months
GENDER_NAME	Gender
DISTRIBUTOR_NAME	Bank Name
CreditLimitAmt	Credit limit
HAS_ESTATEMENT_AGREEMENT_IND	Indicator, direct debit agreement selected ("avtalegiro")
HAS_DIRECT_DEBIT_AGREEMENT_IND	Indicator, e-statement selected ("e-faktura")
DaysFirstUse	Number of days between card is issued and the first time of use
INTEREST_EARNING_LENDING_AMT	Interest earning balance at the month of the campaign
PNRSerial	Digits 7 og 8 in the national identification number
CLOSING_BALANCE_AMT	Total amount printed on Statement
PrevMinimunToPayAmt	Minimum to pay previous month
MINIMUM_TO_PAY_AMT	Minimum to pay at the month of the campaign
BALANCE_AMT	Balance at the end of the month
CASH_BALANCE_AMT	Cash balance at the end of the month
SumAirlineL3	Sum of transactions in given class last 3 months
SumELECTRIC_APPLIANCEL3	Sum of transactions in given class last 3 months

Table 5.1: A description of the variables

Variable Name	Description
SumFOOD_STORES_WAREHOUSEL3	Sum of transactions in given class last 3 months
SumHOTEL_MOTELL3	Sum of transactions in given class last 3 months
SumHARDWAREL3	Sum of transactions in given class last 3 months
SumINTERIOR_FURNISHINGSL3	Sum of transactions in given class last 3 months
SumOTHER_RETAILL3	Sum of transactions in given class last 3 months
SumOTHER_SERVICESL3	Sum of transactions in given class last 3 months
SumOTHER_TRANSPORTL3	Sum of transactions in given class last 3 months
SumRECREATIONL3	Sum of transactions in given class last 3 months
SumRESTAURANTS_BARSL3	Sum of transactions in given class last 3 months
SumSPORTING_TOY_STORESL3	Sum of transactions in given class last 3 months
SumTRAVEL_AGENCIESL3	Sum of transactions in given class last 3 months
SumVEHICLESL3	Sum of transactions in given class last 3 months
SumQuasiCashL3	Sum of transactions easily converted into cash in the last 3 months
SumAirlineL12	Sum of transactions in given class last 12 months
SumELECTRIC_APPLIANCEL12	Sum of transactions in given class last 12 months
SumFOOD_STORES_WAREHOUSEL12	Sum of transactions in given class last 12 months
SumHOTEL_MOTELL12	Sum of transactions in given class last 12 months

Table 5.1: A description of the variables

Variable Name	Description
SumHARDWAREL12	Sum of transactions in given class last 12 months
SumINTERIOR_FURNISHINGSL12	Sum of transactions in given class last 12 months
SumOTHER_RETAILL12	Sum of transactions in given class last 12 months
SumOTHER_SERVICESL12	Sum of transactions in given class last 12 months
SumOTHER_TRANSPORTL12	Sum of transactions in given class last 12 months
SumRECREATIONL12	Sum of transactions in given class last 12 months
SumRESTAURANTS_BARSL12	Sum of transactions in given class last 12 months
SumSPORTING_TOY_STORESL12	Sum of transactions in given class last 12 months
SumTRAVEL_AGENCIESL12	Sum of transactions in given class last 12 months
SumVEHICLESL12	Sum of transactions in given class last 12 months
SumQuasiCashL12	Sum of transactions easily converted into cash in the last 12 months
SUM_of_FirstDunningFlag	Number of months with dunning (”purring”) in the last 12 months
SUM_of_RevolvingFlag	Number of months with where interest is paid in the last 12 months
SUM_of_FullpayerFlag	Number of months where total balance is paid in the last 12 months
SUM_of_CollectionAdviceFlag	Number of months with collection notice (”inkassovarsel”) in the last 12 months
SUM_of_OverdraftFlag	Number of months with overdraft last 12 months
SUM_of_CreditLimitIncreaseFlag	Number of credit limit increases last 12 months

Table 5.1: A description of the variables

Variable Name	Description
SUM_of_CreditLimitDecreaseFlag	Number of credit limit decreases last 12 months
QUASI_CASH	Sum of transactions easily converted into cash in the last month
AIRLINE	Sum of transactions in given class in the last month
rev_per_uti_change_L3M	Change in revolving utilisation (revolving balance divided by credit limit) last 3 months
average_credit_limit_last12	Average credit limit last 12 months
average_revolvingbalance_last12	Average revolving balance last 12 months
avg_rev_bal_L3M	Average revolving balance last 3 months
avg_payment_L3M	Average payment (”innbetaling”) last 3 months
payment_amt_change_L3M	Change in payment amount last 3 months
Segment9Name	Segment name according to separate documentation
Segment23Name	Segment name according to separate documentation
SCORE	Simple risk score according to separate documentation
Distinct_industries	Number of distinct industries where transaction have occurred
DaysSinceCash	Days since cash withdrawal at application date or end of month
DaysSincePurchase	Days since purchase at application date or end of month
DaysSinceTransfer	Days since card to bank account transfer at application date or end of month

Table 5.1: A description of the variables

Variable Name	**Description**
revUtilizationL12	Average revolving balance last 12 months divided by average credit limit last 12 months
AvgRevBalL3onL12	Average revolving balance last 3 months divided by / Average revolving balance last 12 months
PaymentPartofCLL3	Average payment (”innbetaling”) last 3 months divided by average credit limit last 12 months
ResponseInd2	Indicator, applied for credit limit increase

BIC Model summary

```
> summary(BIC)

Call:
glm(formula = ResponseInd2 ~ MonthsSinceAccountCreated + GENDER_NAME +
    CreditLimitAmt + HAS_ESTATEMENT_AGREEMENT_IND + DaysFirstUse +
    INTEREST_EARNING_LENDING_AMT + PrevMinimunToPayAmt + BALANCE_AMT +
    CASH_BALANCE_AMT + SumTRAVEL_AGENCIESL3 + SumQuasiCashL3 +
    SumINTERIOR_FURNISHINGSL12 + SumTRAVEL_AGENCIESL12 + SUM_of_FirstDunningFlag +
    SUM_of_RevolvingFlag + SUM_of_FullpayerFlag + SUM_of_OverdraftFlag +
    SUM_of_CreditLimitIncreaseFlag + SUM_of_CreditLimitDecreaseFlag +
    average_credit_limit_last12 + avg_rev_bal_L3M + SCORE + Distinct_industries +
    revUtilizationL12 + PNRSerial2 + OfferedBeforeFlag + transfercut +
    purchasecut, family = binomial(link = "logit"), data = ntrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9136  -0.3438  -0.2684  -0.2145   8.4904

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -2.176e+00  1.256e-01 -17.323  < 2e-16 ***
MonthsSinceAccountCreated     -7.871e-03  7.960e-04  -9.888  < 2e-16 ***
GENDER_NAMEMann                2.859e-01  2.348e-02  12.176  < 2e-16 ***
CreditLimitAmt                 2.649e-04  3.008e-06  88.066  < 2e-16 ***
HAS_ESTATEMENT_AGREEMENT_IND1  1.540e-01  2.469e-02   6.237 4.45e-10 ***
DaysFirstUse                   1.273e-04  3.314e-05   3.840 0.000123 ***
INTEREST_EARNING_LENDING_AMT  -7.916e-06  1.853e-06  -4.273 1.93e-05 ***
PrevMinimunToPayAmt            1.033e-04  2.390e-05   4.321 1.55e-05 ***
BALANCE_AMT                   -2.773e-05  1.630e-06 -17.010  < 2e-16 ***
CASH_BALANCE_AMT              -7.233e-06  1.353e-06  -5.345 9.04e-08 ***
SumTRAVEL_AGENCIESL3           1.223e-05  2.956e-06   4.138 3.50e-05 ***
SumQuasiCashL3                 1.263e-05  1.994e-06   6.333 2.41e-10 ***
SumINTERIOR_FURNISHINGSL12    -9.325e-06  2.005e-06  -4.650 3.31e-06 ***
SumTRAVEL_AGENCIESL12         -7.483e-06  1.883e-06  -3.975 7.05e-05 ***
SUM_of_FirstDunningFlag       -8.301e-02  2.054e-02  -4.041 5.33e-05 ***
SUM_of_RevolvingFlag          -4.558e-02  6.157e-03  -7.403 1.34e-13 ***
SUM_of_FullpayerFlag          -2.377e-02  6.287e-03  -3.781 0.000156 ***
SUM_of_OverdraftFlag           1.220e-01  2.593e-02   4.705 2.54e-06 ***
SUM_of_CreditLimitIncreaseFlag -8.051e-01  3.389e-02 -23.759  < 2e-16 ***
SUM_of_CreditLimitDecreaseFlag  2.116e+00  8.387e-02  25.233  < 2e-16 ***
average_credit_limit_last12   -2.749e-04  3.170e-06 -86.719  < 2e-16 ***
avg_rev_bal_L3M               -9.490e-06  1.812e-06  -5.238 1.62e-07 ***
SCORE1                        -6.910e-02  1.032e-01  -0.670 0.502997
SCORE2                         6.293e-02  1.041e-01   0.605 0.545371
SCORE3                         2.094e-01  1.082e-01   1.936 0.052840 .
SCORE4                         3.323e-01  1.111e-01   2.991 0.002784 **
SCORE5                         5.026e-01  1.144e-01   4.395 1.11e-05 ***
SCORE6                         7.455e-01  1.275e-01   5.846 5.04e-09 ***
SCORE7                         7.397e-01  1.775e-01   4.168 3.07e-05 ***
Distinct_industries            9.399e-03  2.598e-03   3.618 0.000297 ***
revUtilizationL12              7.092e-01  8.842e-02   8.021 1.05e-15 ***
PNRSerial2TRUE                -3.087e-01  2.800e-02 -11.022  < 2e-16 ***
OfferedBeforeFlagTRUE         -1.998e-01  3.161e-02  -6.321 2.60e-10 ***
transfercut2. >50             -4.107e-01  5.433e-02  -7.558 4.08e-14 ***
transfercut3. Never           -3.471e-01  5.194e-02  -6.684 2.33e-11 ***
```

```
purchasecut2. >50              -2.170e-01  4.170e-02  -5.203 1.96e-07 ***
purchasecut3. Never            -1.342e-02  4.804e-02  -0.279 0.780017
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 86738  on 152698  degrees of freedom
Residual deviance: 60035  on 152662  degrees of freedom
AIC: 60109

Number of Fisher Scoring iterations: 7
```

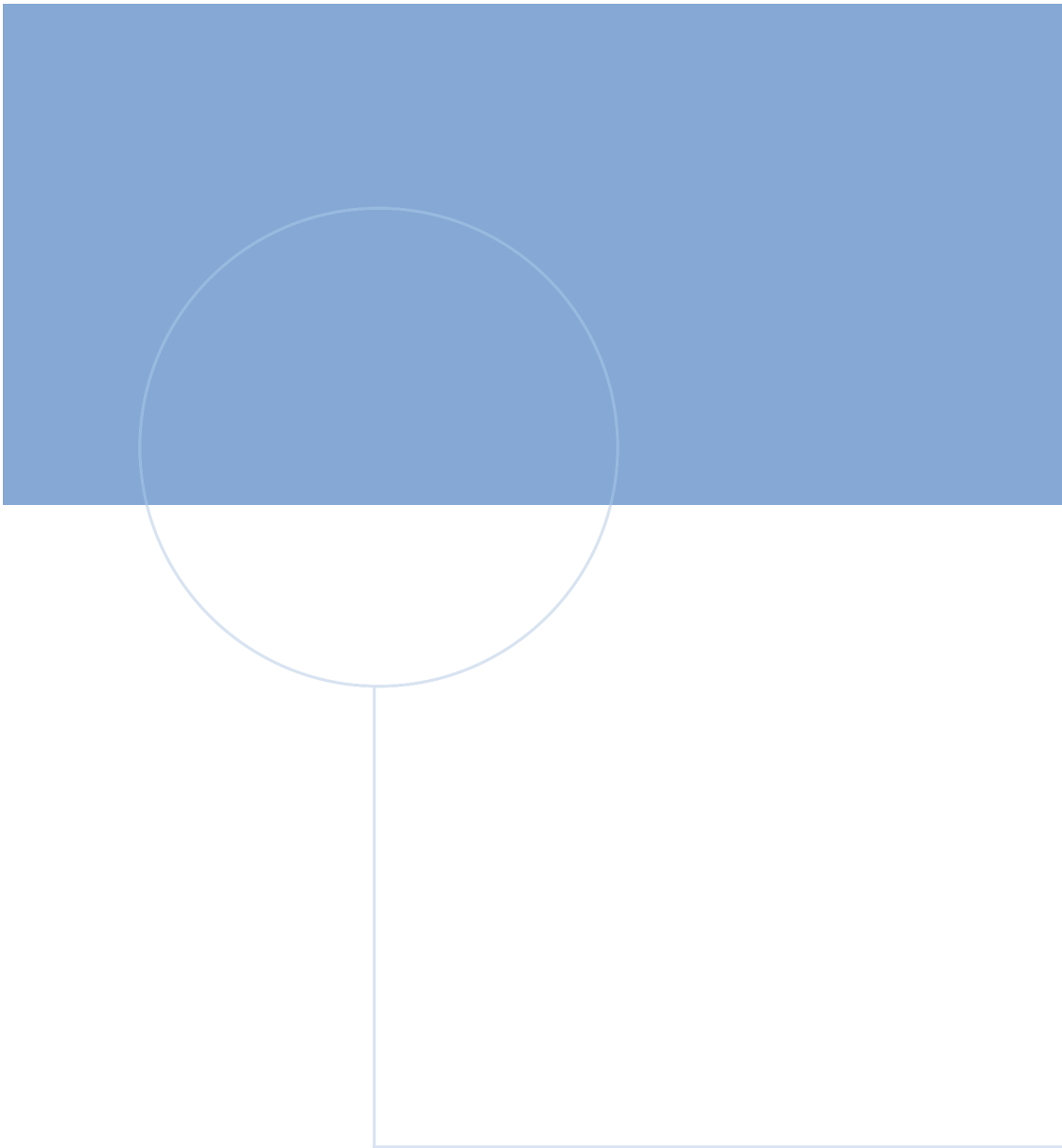
Logit L1 regularization coefficients

Listing 5.1: Coefficients for the regularized logit model. Coefficients shrinked to 0 are marked by a dot.

```
glmnet

152699 samples
    71 predictor
     2 classes: 'FALSE.', 'TRUE.'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 122159, 122159, 122159, 122159, 122160
Resampling results:

  ROC        LiftIndex
  0.8193301  0.8315237

Tuning parameter 'alpha' was held constant at a value of 1
Tuning parameter 'lambda' was held constant at a value of 0.002
> coef(l1$finalModel, 0.002)
124 x 1 sparse Matrix of class "dgCMatrix"
                                                                      1
(Intercept)                                                -2.200871e+00
CustomerAge                                                -1.677492e-03
MonthsSinceAccountCreated                                  -4.076470e-03
GENDER_NAMEMann                                             2.275461e-01
DISTRIBUTOR_NAMESpareBank 1 BV                              .
DISTRIBUTOR_NAMESpareBank 1 Gudbrandsdal                    .
DISTRIBUTOR_NAMESpareBank 1 Hallingdal Valdres              .
DISTRIBUTOR_NAMESpareBank 1 Lom og Skj k                   -1.296901e-02
DISTRIBUTOR_NAMESpareBank 1 Modum                           .
DISTRIBUTOR_NAMESpareBank 1 Nord-Norge                      .
DISTRIBUTOR_NAMESpareBank 1 Nordvest                        .
DISTRIBUTOR_NAMESpareBank 1 N tter y -T nsberg                 .
DISTRIBUTOR_NAMESpareBank 1 Oslo Akershus                   .
DISTRIBUTOR_NAMESpareBank 1  stfold  Akershus                .
DISTRIBUTOR_NAMESpareBank 1  stlandet                        .
DISTRIBUTOR_NAMESpareBank 1 Ringerike Hadeland              .
DISTRIBUTOR_NAMESpareBank 1 SMN                            -1.432327e-01
DISTRIBUTOR_NAMESpareBank 1 S re Sunnm re                     .
DISTRIBUTOR_NAMESpareBank 1 SR-Bank                         4.096410e-04
DISTRIBUTOR_NAMESpareBank 1 Telemark                        .
CreditLimitAmt                                              1.942820e-04
HAS_ESTATEMENT_AGREEMENT_IND1                               2.661259e-02
HAS_DIRECT_DEBIT_AGREEMENT_IND1                             1.656146e-02
DaysFirstUse                                                .
INTEREST_EARNING_LENDING_AMT                                .
CLOSING_BALANCE_AMT                                         .
PrevMinimunToPayAmt                                         .
MINIMUM_TO_PAY_AMT                                         -7.638863e-05
BALANCE_AMT                                                -1.967169e-05
CASH_BALANCE_AMT                                           -5.219229e-06
SumAirlineL3                                                .
SumELECTRIC_APPLIANCEL3                                     .
SumFOOD_STORES_WAREHOUSEL3                                  .
```

```
SumHOTEL_MOTELL3                                    .
SumHARDWAREL3                                       .
SumINTERIOR_FURNISHINGSL3                           .
SumOTHER_RETAILL3                                   .
SumOTHER_SERVICESL3                                 .
SumOTHER_TRANSPORTL3                                .
SumRECREATIONL3                                     .
SumRESTAURANTS_BARSL3                               .
SumSPORTING_TOY_STORESL3                            .
SumTRAVEL_AGENCIESL3                                .
SumVEHICLESL3                                       .
SumQuasiCashL3                                      5.943045e-06
SumAirlineL12                                       .
SumELECTRIC_APPLIANCEL12                            .
SumFOOD_STORES_WAREHOUSEL12                         .
SumHOTEL_MOTELL12                                   .
SumHARDWAREL12                                      .
SumINTERIOR_FURNISHINGSL12                          -1.496164e-06
SumOTHER_RETAILL12                                  .
SumOTHER_SERVICESL12                                .
SumOTHER_TRANSPORTL12                               .
SumRECREATIONL12                                    .
SumRESTAURANTS_BARSL12                              .
SumSPORTING_TOY_STORESL12                           -5.820787e-07
SumTRAVEL_AGENCIESL12                               .
SumVEHICLESL12                                      -1.687195e-07
SumQuasiCashL12                                     .
SUM_of_FirstDunningFlag                             .
SUM_of_RevolvingFlag                                .
SUM_of_FullpayerFlag                                -5.236826e-04
SUM_of_CollectionAdviceFlag                         .
SUM_of_OverdraftFlag                                8.371503e-02
SUM_of_CreditLimitIncreaseFlag                      -2.902610e-01
SUM_of_CreditLimitDecreaseFlag                      1.466597e+00
QUASI_CASH                                          .
AIRLINE                                             .
rev_per_uti_change_L3M                              .
average_credit_limit_last12                         -2.038849e-04
average_revolvingbalance_last12                     .
avg_rev_bal_L3M                                     .
avg_payment_L3M                                     .
payment_amt_change_L3M                              .
Segment9NameEMOB - Not active last 6 mths           .
Segment9NameLast active 4-6 mths ago                .
Segment9NameLast active 7-12 mths ago               .
Segment9NameNot active in last 12 mths              3.911967e-01
Segment9NameOccasional Revolver                     .
Segment9NameRevolved only                           -3.851676e-03
Segment9NameRevolver                                .
Segment9NameTransactor                              .
Segment23NameActive 7-12 mths ago                   .
Segment23NameEMOB - Active in last 6 mths           8.974998e-02
Segment23NameEMOB - Not active last 6 mths          .
Segment23NameNot active in last 12 mths             2.138327e-03
Segment23NameOcc. Revolver: Good spend & low headroom       .
Segment23NameOcc. Revolver: Good spend & poor revolve freq  .
Segment23NameOcc. Revolver: Good spend/Good headroom        .
Segment23NameOcc. Revolver: Poor spend & borrowing          1.700466e-02
```

Segment23NameOcc. Revolver: Poor spend/Low headroom	1.507128e-01
Segment23NameRevolved Only: Balance at risk	.
Segment23NameRevolved Only: Balance not at risk yet	.
Segment23NameRevolved Only: Paid off	.
Segment23NameRevolver: Good spend & large headroom	.
Segment23NameRevolver: Good spend/Good headroom	.
Segment23NameRevolver: Good spend/low headroom	.
Segment23NameRevolver: Low spend & Good/large headroom	-4.948092e-02
Segment23NameRevolver: Poor spend/low headroom	.
Segment23NameTransactor: Everyday Expences	.
Segment23NameTransactor: Front of wallet	.
Segment23NameTransactor: Low User	.
Segment23NameTransactor: Occasional Spender	.
Segment23NameTransactor: Regular Spender	.
SCORE1	-3.107099e-02
SCORE2	.
SCORE3	.
SCORE4	7.162677e-02
SCORE5	1.757060e-01
SCORE6	4.126661e-01
SCORE7	5.466988e-03
Distinct_industries	.
revUtilizationL12	.
AvgRevBalL3onL12	.
PaymentPartofCLL3	.
PNRSerial2TRUE	-2.676861e-01
OfferedBeforeFlagTRUE	-1.890706e-01
cashcut2. >50	.
cashcut3. Never	.
transfercut2. >50	-1.502379e-01
transfercut3. Never	-2.404563e-01
purchasecut2. >50	-7.257770e-02
purchasecut3. Never	.



NTNU
Norwegian University of
Science and Technology