NTNU
Norwegian University of
Science and Technology

Predicting Recovery Rates of Defaulted Credit Card Accounts

Anine Harto

Master of Science in Physics and Mathematics
Submission date: June 2018
Supervisor: John Sølve Tyssedal, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU), and concludes my studies in the field of applied mathematics. The work was conducted during the spring semester of 2018 at the Department of Mathematical Science, in cooperation with SpareBank 1 Kredittkort with Christian Meland as an external supervisor.

The project serves as a contribution to the ongoing work of modelling credit card risk behaviour. It is assumed that the reader is familiar with some statistical modelling, and a background in computer science or mathematics is preferred.

I would like to express my gratitude for the support and counselling provided by my supervisor John Sølve Tyssedal. Furthermore, many thanks to Christian Meland, Hans Bystrøm and Jens Morten Nilsen at Sparebank 1 for their guidance and the opportunity of writing this thesis. It is with ambivalent feelings that I am writing these final sentences. My time at NTNU has been the most extraordinary years of my life.

Trondheim, 13.06.18
Anine Harto

Abstract

Despite the rapidly evolving payment options available to consumers, credit cards still remain the leading payment method and are also one of the most lucrative forms of banking. As there will always exist borrowers who default on their obligations, some loss must be expected. However, recognizing which debtors that are most likely not to repay their debt in full after default is crucial when estimating these losses. Additionally, obtaining a model that accurately predicts the fraction of debt a customer is able to restore can be a source of great revenue. Instead of using resources on collecting debts we predict will not be restored, these could be assigned to cases where the outcome is more uncertain. Thus, helping debtors to recover from default is beneficial both for the consumer and the credit card distributor.

This thesis presents three risk models, all with the aim of predicting how much of the debt a defaulted customer is able to repay. Using a dataset consisting of credit card information registered during the period August 2015 to November 2017, logistic regression models, support vector machines and fuzzy clustering models were constructed. The models were also built with the aim of detecting the behaviour of high-risk customers.

The main contribution of this thesis has been to illustrate how conventional, and some unconventional, statistical methods can be used to reveal trends and perform inference on data. As the use of machine learning algorithms and black-box methods thrive, it can be difficult to pinpoint why a given algorithm works/does not work as well as being able to interpret the models. Despite none of the models proved themselves as optimal, several contributed to a greater understanding of what separates low-risk from high-risk customers. An improvement of the model would be to include other types of personal information gathered from several databases. Information regarding health care status, insurance purchases, social media activity, shopping habits etc. would not only improve the models in this thesis but all modelling involving human behaviour.

Sammendrag

Til tross for den stadige utviklingen av ulike betalingsløsninger tilgjengelig for forbrukere, er kredittkort fortsatt den ledende betalingsmetoden, i tillegg til å være en av de mest lukrative tjenestene for banker. Siden det alltid vil eksistere låntakere som ikke klarer å oppfylle betalingskravene, vil det alltid forventes noe tap. Ved å gjenkjenne hvilke låntakere som har størst sannsynlighet for å ikke klare å tilbakebetale deres respektive gjeld etter å ha bli sendt til mislighold, er avgjørende i estimeringen av forventet tap. Å anskaffe seg en modell som til et tilfredsstillende nivå klarer å predikere andelen gjeld en kunde klarer å betale tilbake, kan være en stor inntektskilde. Istedenfor å bruke ressurser på å samle gjeld vi allerede har predikert at vi ikke klarer å få tilbake, kan disse ressursene bli fordelt til tilfellene hvor utfallet er mer uvisst. På den måten kan vi hjelpe låntakere å komme tilbake fra mislighold, noe som både er gunstig for forbruker og bank.

Denne avhandlingen vil presentere tre risikomodeller der alle har som mål å predikere hvor stor andel en kunde som er sendt til mislighold klarer å tilbakebetale. Ved å bruke data bestående av kredittkortinformasjon registrert i perioden august 2015 til november 2017, har logistiske regresjonsmodeller, support vector machines og fuzzy clustering modeller blitt konstruert. Disse modellene ble også bygget med et mål om å gjenkjenne oppførsel som tyder på høy risiko hos kunder.

Hovedbidraget til denne avhandlingen har vært å illustrere hvordan konvensjonelle, og noen ukonvensjonelle, statistiske metoder kan bli brukt til å avsløre trender og utføre inferens på data. Ettersom bruken av maskinlæringsalgoritmer og black-box-metoder blomstrer for fullt, kan det være vanskelig å fastslå hvorfor en gitt algoritme virker eller ikke virker, i tillegg til å tolke modellene. Til tross for at ingen av modellene viste seg å være optimale, har flere bidratt til en økt forståelse for hva som skiller lavrisiko- fra høyrisikokunder. En forbedring av modellene ville vært å inkludere annen informasjon samlet fra flere databaser. Informasjon som inneholder helsestatus, forsikringskjøp, aktivitet på sosiale medier, kjøpshistorikk osv. ville ikke bare forbedret modellene i denne avhandlingen, men også all modellering som involverer menneskelig oppførsel.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

One might think that the credit card industry is rather new, but it actually dates back to the early 1800s [1]. Even though plastic cards were not in use back then, merchants and financial intermediaries did extend credit on durable goods. Already in the early 1900s larger hotels and department stores began to distribute paper cards to their most devoted customers.

Today, with the rapidly evolving payment options available to consumers, credit cards still remain the leading payment method. Actually, according to the 2016 U.S Consumer Payment Study [2], for the first time in years has credit taken over the top spot as the overall preferred way to pay, replacing debit. So, despite the growth of mobile payments, traditional payment methods still remain highly relevant.

One main aspect of the credit card industry is the loss of interest as a result of customers becoming severely delinquent on their credit card payment. Some loss is always expected as there always exist borrowers who default on their obligations. Financial institutions view these losses as a cost component of doing business, and so they are managed through several means, including interest rates and pricing of credit exposures. However, the exact loss observed in a particular year vary between years even though the quality of the portfolio is assumed consistent over time. Figure 1.1 illustrates how the realized loss can be divided into expected loss (EL) and unexpected loss (UL).

Unexpected losses, which are losses above the expected levels, are known to occur now and then. It is, however, difficult to predict their timing and severity. Financial institutions are therefore required to hold parts of their capital as a buffer to protect the debtors against these unexpected losses. The Basel Accords [3], three sets of banking regulations (Basel I, II, III) set by the Basel Committee on Bank Supervision, recommend banks to develop their internal credit risk model for expected loss. One approach adopted for Basel II looks at the probability of loss exceeding the unexpected level by means of a stochastic portfolio model. The frequency curve on the right-hand side of Figure 1.1 describes the probability of losses of a certain magnitude.

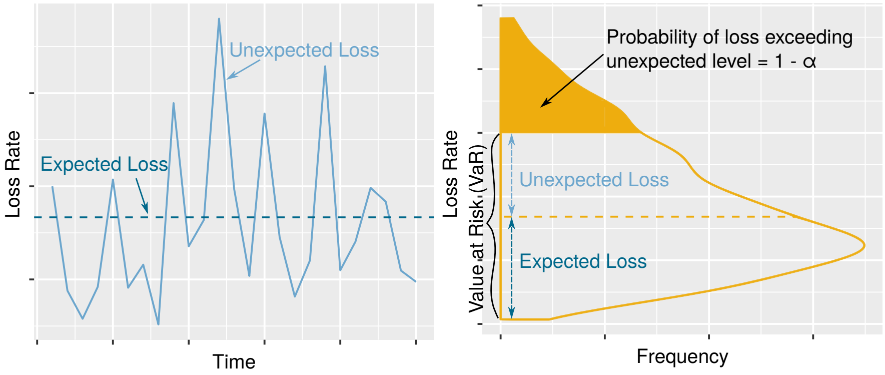


Figure 1.1: Variation in realized losses over time (left) generates a distribution of losses (right). The dashed line represents the expected loss, and the spikes above the dashed line are known as unexpected loss.

We see that small losses around the EL occur more frequently than large losses. The probability that a bank will not be able to meet its own credit obligation, i.e the losses exceed the unexpected level, equals the coloured area. 1 minus this probability is the confidence level, denoted α. The corresponding threshold is called the Value-at-Risk (VaR) for this confidence level. Hence, letting the buffer size be the difference between EL and VaR when EL is covered by provisions and revenues, gives a probability of a bank remaining solvent over a one-year horizon equal to the confidence level. Here, the EL is considered from a portfolio perspective. We can also view it as a sum of different components, namely

$$\text{EL} = \text{PD} \times \text{EAD} \times \text{LGD},$$

where the Probability of Default (PD) is the average percentage of borrowers who default during the course of one year, the Exposure at Default (EAD) is the estimated outstanding amount from a defaulted customer, and Loss Given Default (LGD) is the ratio of outstanding debt the bank might lose if a customer defaults. These are the essential risk parameters to be estimated in the Advanced Internal Rating Based (AIRB) approach proposed in Basel II.

In this paper, our main aim is to estimate the value of LGD, or more precisely the recovery rate (RR) given as the complementary event of LGD. That is, we want to estimate the percentage of outstanding debt a defaulted customer is able to restore. A defaulted customer is able to change their status back to normal if they are able to restore parts of their debt, meaning their RR is above zero. The RR is defined as 1 representing total recovery, and 0 representing a total loss.

1.2 Motivation

Numerous approaches have been made in order to accurately predict the RR of a defaulted borrower. Hwang et al. (2014) [4] propose a Two-stage Probit Model (TPM) to predict RR due to its ordinal nature: total loss, total recovery, and lying between the two extremes. The two-stages consist of first utilizing the ordered probit model to predict which of the three categories the account belongs to. Next, to predict the accurate recovery rate for accounts classified in between the two extremes, the probit transformation regression is used. The analysis is based on real data, and the results indicate that the TPM performs better than its competitive alternatives such as simple probit transformation regression, the mixture of Bernoulli and beta random variables and the decision tree model.

Moore (2017) [5] performed a case study from a debt collection agency in London assuming beta distributed recovery rates. The most important aspect of the research is the interpretation of predictors. Regardless of a less than optimal fit, the model reveals a higher RR for older borrowers, women opposed to men, borrowers who were homeowners and for the cases when the debt was less than £100.

Yao et al. (2017) [6] predict RR through the incorporation of least squares Support Vector Machine (SVM) techniques into a two-stage modelling framework. Similarly to Hwang et. al (2014), this model requires a classification step discriminating the cases with RR equal to either 0 or 1, in addition to a regression step that estimates the RR for the cases in between. The result indicates that the SVM is preferred to a logistic regression using an out-of-time sample. However, modelling on the whole sample does not give the support vector machine any advantage compared with other techniques within the two-stage modelling framework.

The introduction of the Basel Accords emphasized the importance of the estimation of LGD to the banking world. First, recognizing the factors which affect the RR is extremely important in calculating the LGD for debtors. This is in return crucial in the estimation of the expected losses when determining how much buffer the bank should hold. Second, learning which debtors are more likely to pay after they have defaulted means that debt collectors can shift their focus towards the debtors we do not know for certain the outcome of. Thus, it is possible to help the debtors which initially did not have the intention or means to pay down their debt. Third, identifying the debtors that are at greater risk of not paying, can help with a better pricing of debt.

1.3 Approach

The problem at hand is to implement predictive models to decide if a defaulted customer will be able to pay the minimum amount required in order to return back to status *normal* on their payments. These models will be built using mainly three statistical methods with the data provided by SpareBank 1: logistic regression, support vector machine and fuzzy clustering. In a predictive analysis, logistic regression has for long been the standard method when the outcome is binary. As for support vector machines, they have some advantages over more classical approaches such as logistic regression. SVMs are able to handle non-linear feature interactions in addition to large feature spaces. Another approach of classification is the unsupervised method of clustering where data of similar types are assigned to the same cluster. Fuzzy clustering offers the measure of the degree of membership in $[0, 1]$, yielding great flexibility in the sense that data points can belong to more than one cluster.

Chapter 2 introduces the dataset used for modelling, both the included and generated variables, as well as our response of interest. Additionally, some visualizations are going to be presented to give some impression on how some variables are related. Chapter 3 presents the necessary theory behind the fitting of the models, as well as validity - and predictive measures. Chapter 4 gives the results obtained from model fitting, and an analysis with respect to the goodness of fit and predictive power. Finally, Chapter 5 concludes the thesis with some closing remarks and some ideas for further work.

Chapter 2

Data

The dataset used is provided by Sparebank 1, an alliance of 16 different Norwegian banks. The dataset consists of credit card information from 40639 distinct collection cases through the period of August 2015 to November 2017. The dataset includes information regarding transactions, application records and insolvency records. Personal information, however, such as names, addresses, phone number and social security number, is removed or modified due to the sensitive nature of credit card analyses. The variables at hand will first be presented, before defining our response of interest. Finally, some illustration of the dataset will be included.

2.1 Variables

Table A.1 in appendix A presents a complete list of the available variables, both categorical and continuous, with explanation. The variables can be divided into transactional, account- and application variables.

Transactional variables are mainly observations for each individual credit card account reported at the end of each month. This includes observations from the invoice that is due during the current month, such as the accumulated interest at the last day of the current month. Other transactional variables are aggregated data through the month, and includes variables such as the average balance during the current month and closing balance. Flags raised during the month are also included here. Such flags could be if the account is late for a payment, or if there has been a change in the credit limit during the month. Finally, transactional variables include data describing the account purchases, transactions and cash withdrawals during the first 12 months.

Static data applicable for all months are denoted account variables, and consists of the customer's age, gender and credit limit, among others. Additionally an accounts insolvency history is reported here, and includes variables such as the amount of times the account has been sent to collection previously, and the number of received collection warnings.

Application variables are the ones given in the customer application, and includes variables such as gross income and total amount of mortgages.

2.2 Response

One important complication when constructing models to predict LGD is the common occurrence of its distribution being bimodal with modes at 0 and 1, which is observed to the left of Figure 2.1. This motivates the questions of how to transform the covariates and which model algorithm to construct. As mentioned in chapter 1 several approaches have been considered in the search of a solution to this problem. One different approach would be to define LGD as binary instead of continuous on the bounded interval $[0, 1]$. This is done by defining LGD as 0 if the borrower manages to return back to *normal* independent on how much money is refunded, and 1 if the borrower remains defaulted. Rather than focusing on how much debt is lost, we choose the response of interest to be the RR. The recovery rate is defined as the ratio of outstanding debt a defaulted customer is able to pay back, and is therefore simply the complementary event of LGD. Hence, our response of interest is

$$\text{RR} = 1 - \text{LGD} = \begin{cases} 1, & \text{if customer returns to status } \textit{normal} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

One way to look at this problem is to divide all defaulted customers into three categories: The ones we know for certain will pay the minimum amount required, the ones we know will not, and the ones who could end up in either of the former categories. There is no gain in using resources on customer we know the outcome of. The main interest is therefore the customers who, with some assistance, might be able to pay. Turning such customers will be a source of great revenue. Utilizing this definition of the recovery rate, our data is divided as presented to the right in Figure 2.1.

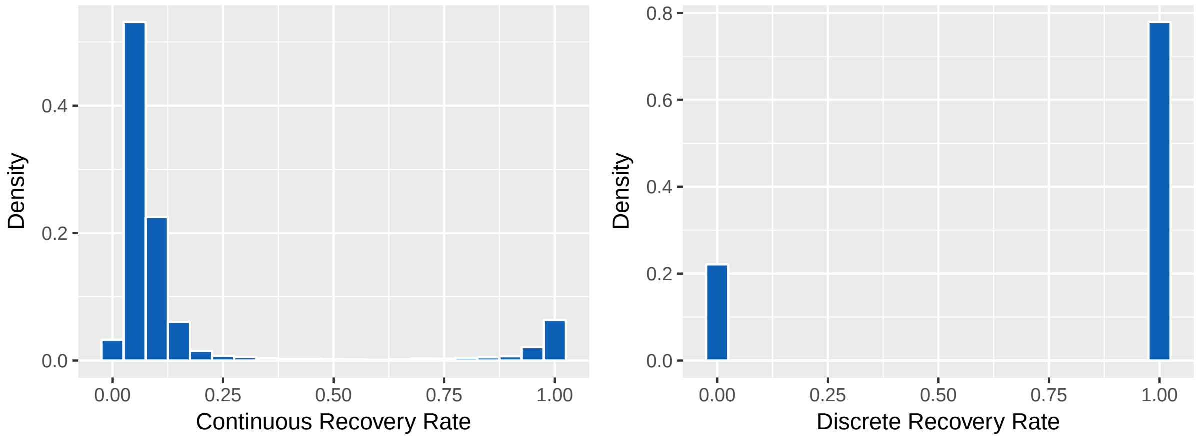


Figure 2.1: Distribution of the recovery rate using both the continuous definition (left) and discrete definition (right)

Henceforth, accounts with $\text{RR} = 1$ are called low risk or recovering accounts, while accounts with $\text{RR} = 0$ are called high-risk or non-recovering accounts.

2.3 Visualization of Data

In order to obtain some notion of how recovered accounts differ from non-recovered accounts, we will take a look at the dataset. Figure 2.2 shows the development of the closing balance divided by credit limit each month for a subset of accounts younger than 6 months. The plots indicates no particular spending pattern for recovered accounts versus non-recovered accounts. One can notice that several non-recovered accounts exhibits the behaviour of having a closing balance close to the credit limit each month. This behaviour exists for recovered accounts as well, but possibly not to the same extent as the variability is much higher here.

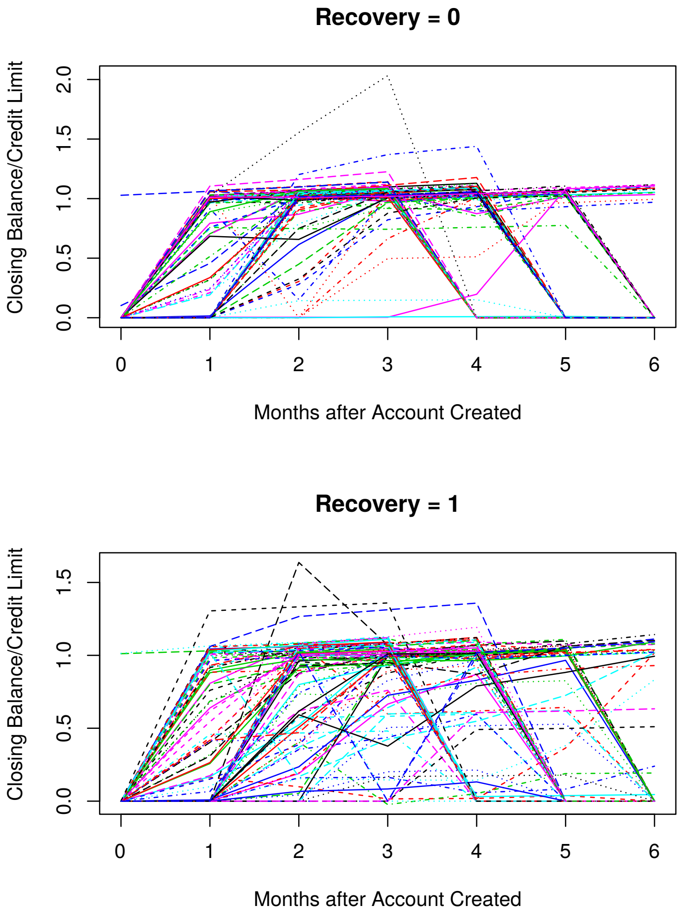


Figure 2.2: Spending pattern for non-recovered (top) and recovered (bottom) accounts younger than 6 months

Figure 2.3 reveals a linearly increasing effect on the average RR when the number of times the account has defaulted previously lay between 0 and 4. Thereafter it flattens out. The same pattern can be detected for the number of times a warning of collection has been sent to the account. Both of these observations imply that customers who frequently are recovering from collection or exhibit unwanted behaviour, have a higher probability of recovering later. It is the first timers who have the lowest recovery rate. Considering the age of the account and customer, one would expect that older ones have a higher capability of recovering. For really low values of either variable this effect is evident. Once the age of the account exceeds 10 months this influence is still positive, however not as dominant. Similarly for the customer's age, this effect is not observable for customers older than 23 years.

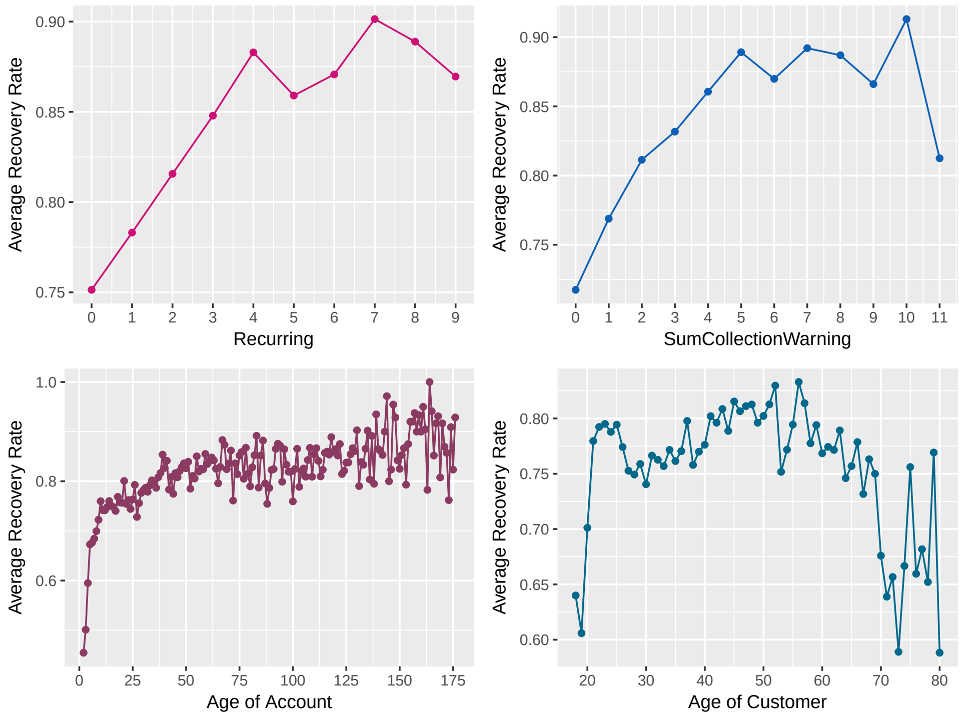


Figure 2.3: Different explanatory variables plotted against the average recovery rate

Chapter 3

Statistical Models and Methods

This section provides the necessary theory for constructing and analyzing predictive models for the data described in chapter 2. The concepts of logistic regression, support vector machine and fuzzy clustering are introduced, in addition to presenting common methods for model diagnostics for each of the models.

3.1 Logistic Regression

3.1.1 The Logistic Model

Rather than modeling the response Y directly, logistic regression models the probability of Y belonging to a particular category [7, 8].

We assume that data on N objects are given in the form $(y_i, x_{i1}, ..., x_{ik}), i = 1, ..., N$, with the binary response y belonging to two categories, coded by 0 and 1, and covariates denoted by $x_1, ..., x_k$. Thus, we have

$$Y_i \sim \text{Bin}(n_i = 1, \pi_i).$$

The aim of regression with binary responses is to model the expected value, i.e the conditional probability

$$\text{E}(Y_i|\mathbf{x}) = \text{P}(Y_i = 1|\mathbf{x}) = \pi_i.$$

In this specification, the response variables are assumed (conditionally) independent. To model the relationship between π_i and $\mathbf{x}$, we introduce the linear predictor given as

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

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_k)^T$ and $\mathbf{x}_i^T = (1, x_{i1}, ..., x_{ik})$, but since the probability π_i must lie in the interval [0,1], restrictions on $\boldsymbol{\beta}$ are required. These are problematic to handle in the estimation process, and is the reason why the probability π_i is combined with the linear predictor η_i through the relationship

$$\pi_i = h(\eta_i) = h(\beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik}), \tag{3.1}$$

with h given as a strictly monotonically increasing cumulative distribution function on the real line. This ensures $h(\eta_i) \in [0, 1]$, and (3.1) can always be expressed in the form

$$\eta_i = g(\pi_i) = h^{-1}(\pi_i),$$

where g is known as the link function. Choosing the logistic distribution function

$$h(\eta_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

yields the logit model

$$\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} = \frac{\exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik})}. \tag{3.2}$$

Using the link function $g(\pi_i) = \log(\pi_i/(1 - \pi_i))$, known as the canonical link, yields

$$\log\Big(\frac{\pi_i}{1 - \pi_i}\Big) = \log\Big(\frac{\mathrm{P}(Y_i = 1)}{1 - \mathrm{P}(Y_i = 1)}\Big) = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik}, \tag{3.3}$$

or alternatively

$$\frac{\pi_i}{1 - \pi_i} = \frac{\mathrm{P}(Y_i = 1)}{\mathrm{P}(Y_i = 0)} = \exp(\beta_0)\exp(\beta_1 x_{i1}) \cdot ... \cdot \exp(\beta_k x_{ik}), \tag{3.4}$$

where the left hand side of (3.4) is referred to as the odds. Therefore, the interpretation of the estimates changes compared to a linear regression model. We obtain a multiplicative model for the odds, where a unit increase of the value x_{i1} leads to a multiplication of the odds by the factor $\exp(\beta_1)$. This implies a positive effect if $\beta_1 > 0$, a negative effect if $\beta_1 < 0$, and no change if $\beta_1 = 0$. Although a non-linear least squares approach could be made, the more general method of maximum likelihood estimation (MLE) is preferred, as it has better statistical properties.

3.1.2 Estimation of the Regression Coefficients

The coefficients in (3.3) are unknown, and must be estimated based on some available training data. The basic intuition behind the use of maximum likelihood in logistic regression is the desire for obtaining estimates for $\boldsymbol{\beta}$ which ensures a predicted response as close to the observed response y_i as possible. The likelihood function is given as

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{N} f(y_i|\boldsymbol{\beta}) = \prod_{i=1}^{N} \pi_i^{y_i}(1 - \pi_i)^{1-y_i}, \tag{3.5}$$

which depends on $\boldsymbol{\beta}$ through (3.2). Using (3.5) we can find the expression of the log-likelihood

$$\begin{aligned}\ell(\boldsymbol{\beta}) = \log(L(\boldsymbol{\beta})) &= \sum_{i=1}^{N} \ell_i(\boldsymbol{\beta}) \\ &= \sum_{i=1}^{N} \Big[y_i \log(\pi_i) + (1 - y_i)\log(1 - \pi_i) \Big] \\ &= \sum_{i=1}^{N} \Big[y_i \mathbf{x}_i^T \boldsymbol{\beta} - \log(1 + e^{\mathbf{x}_i^T \boldsymbol{\beta}}) \Big].\end{aligned}$$

To maximize the log-likelihood, we set its derivative to zero, which yields the score equations

$$\mathbf{s}(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{N} \mathbf{x}_i (y_i - \pi_i) = 0, \tag{3.6}$$

which are $p+1$ non-linear equations in $\boldsymbol{\beta}$. To solve the score equations (3.6), we use the Newton-Raphson algorithm, which starts with expanding $s(\boldsymbol{\beta})$ in a first-order Taylor series around some chosen reference value $\boldsymbol{\beta}^{(0)}$

$$\mathbf{s}(\boldsymbol{\beta}) \approx \mathbf{s}(\boldsymbol{\beta}^{(0)}) - (\boldsymbol{\beta} - \boldsymbol{\beta}^{(0)})\mathbf{H}(\boldsymbol{\beta}^{(0)}), \tag{3.7}$$

where $\mathbf{H}(\boldsymbol{\beta})$ is denoted the negative Hessian matrix or observed Fisher information matrix, given as

$$\mathbf{H}(\boldsymbol{\beta}) = -\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T \pi_i (1 - \pi_i).$$

Inserting $\mathbf{s}(\boldsymbol{\beta}) = 0$ into (3.7) and solving for $\boldsymbol{\beta}$, we obtain

$$\boldsymbol{\beta} = \boldsymbol{\beta}^{(0)} + \mathbf{H}(\boldsymbol{\beta}^{(0)})^{-1} \mathbf{s}(\boldsymbol{\beta}^{(0)}), \tag{3.8}$$

If we start with some value $\boldsymbol{\beta}^{(0)}$ and find a new value $\boldsymbol{\beta}^{(1)}$ by applying equation (3.8), and then continue applying this equation until convergence we obtain the Newton-Raphson method:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathbf{H}(\boldsymbol{\beta}^{(t)})^{-1} \mathbf{s}(\boldsymbol{\beta}^{(t)}),$$

staring with $\boldsymbol{\beta}^{(\mathbf{0})}$ often equal to zero. When using the canonical link we have that $\mathbf{H}(\boldsymbol{\beta}) = \mathbf{F}(\boldsymbol{\beta})$, i.e the observed fisher information matrix equals the expected fisher information matrix given as

$$\mathbf{F}(\boldsymbol{\beta}) = -\mathrm{E}\Big[\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\Big]. \tag{3.9}$$

Hence, the Newton-Raphson method corresponds here to the Fisher scoring algorithm. Typically the algorithm does converge as the log-likelihood is concave, but overshooting can occur.

For a sufficiently large sample size N, $\hat{\boldsymbol{\beta}}$ obtains an approximate normal distribution

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

where **F** is defined as in (3.9). Hence, significance tests through the standard normal distribution can be performed.

3.1.3 Model Selection

Model selection for logistic regression faces the same problems as for ordinary regression. As the number of explanatory variables increases, the selection process becomes more difficult due to the rapid increase in possible effects and interactions. There are mainly two goals: the model should be complex enough to fit the data well, while also simple to interpret. In other words, smoothing is preferred over overfitting the data.

Two well known procedures for deciding which variables to include are forward selection and backward elimination. Many statisticians prefer backward elimination, which begins with a complex model and sequentially removes terms. At each step it selects the term for which its removal has the least damaging effect on the model, e.g the largest p-value. The process will stop once a removal would lead to a significantly poorer fit of the model.

One other criteria besides the p-value, is the Akaike information criterion, AIC, which judges a model by how close its fitted values tend to be to the true values. Even though a simple model may be further away from the true model than a complex one, it is often preferred as it tends to provide better estimates of certain characteristics of the true model. The criterion is defined as

$$\text{AIC} = -2(\ell - p),$$

where ℓ is the maximized log-likelihood, and $p = k + 1$ is the number of parameters of the model. Hence, another approach for deciding which terms to include is to do a backward elimination, and sequentially remove the term which gives the lowest AIC. The process is terminated once removing a term does not improve the AIC.

By only including a subset of the predictors, subset selection procedures yields a model that is easier to interpret and has possibly a lower prediction error than a full model. However, as the process of including or excluding variables is discrete, the predictor estimates often exhibit high variance, which in turn increases the prediction error. Another alternative is therefore to fit a model containing all p predictors by utilizing a technique that regularizes the coefficient estimates, meaning that they are shrunken towards zero. Shrinkage methods are more continuous, and will therefore not experience such high variability, but at the cost of a small bias in the estimates. The two most common shrinking approaches are ridge regression and the lasso.

We recall from section 3.1.2 that the estimated value for $\boldsymbol{\beta}$ is found by maximizing the log-likelihood described in (3.8). Ridge logistic regression [9] is very similar to this fitting procedure, and is obtained by maximizing the log-likelihood function with an added penalizing parameter applied to all the coefficients except the intercept

$$\begin{aligned}\ell_\lambda^R(\boldsymbol{\beta}) &= \frac{1}{N}\sum_{i=1}^{N}\Big[y_i\mathbf{x}_i^T\boldsymbol{\beta} - \log(1 + e^{\mathbf{x}_i^T\boldsymbol{\beta}})\Big] - \lambda\sum_{j=1}^{p}\beta_j^2 \\ &= \frac{1}{N}\ell(\boldsymbol{\beta}) - \lambda\|\boldsymbol{\beta}\|_2^2,\end{aligned} \tag{3.10}$$

where the log-likelihood is scaled with its sample size N such that values of λ are comparable for different sample sizes [10]. This problem statement is equivalent to

$$\min_{\boldsymbol{\beta}} \quad -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i\mathbf{x}_i^T\boldsymbol{\beta} - \log(1 + e^{\mathbf{x}_i^T\boldsymbol{\beta}})\Big] \quad \text{subject to} \quad \lambda\sum_{j=1}^{p}\beta_j^2 \leq k.$$

The tuning parameter $\lambda > 0$ is to be determined separately. Ridge regression addresses the problem of correlated predictors, as their likelihood estimates can become poorly determined and exhibit high variance. By imposing a size constraint on the coefficients, captured in λ, this issue is alleviated. The tuning parameter λ serves as a weight of the penalization. When $\lambda = 0$, there is no penalization present, and the solution will be the ordinary MLE. However, when $\lambda \to \infty$, the impact of the shrinkage penalty will become large and all the coefficient estimates will tend to zero. The estimates are therefore dependent on the value of λ, such that selecting an optimal value is critical. Also note that the estimates are not equivariant under input scaling, and it is therefore common to standardize the input before solving (3.10).

Ridge regression has one major disadvantage. Unlike forward selection and backward elimination, which selects a subset of of the predictors, ridge regression will always include all p covariates. One will therefore never obtain a parsimonious model when utilizing ridge regression. The penalty term in (3.10) will shrink the coefficients, but not let them be equal to zero. This might not have a great effect on prediction, but it can pose a challenge in interpreting models where p is large. Therefore, the lasso method is introduced. The lasso coefficients maximize the quantity

$$\begin{aligned}\ell_\lambda^L(\boldsymbol{\beta}) &= \frac{1}{N}\sum_{i=1}^{N}\Big[y_i\mathbf{x}_i^T\boldsymbol{\beta} - \log(1 + e^{\mathbf{x}_i^T\boldsymbol{\beta}})\Big] - \lambda\sum_{j=1}^{p}|\beta_j| \\ &= \frac{1}{N}\ell(\boldsymbol{\beta}) - \lambda\|\boldsymbol{\beta}\|_1\end{aligned} \tag{3.11}$$

equivalent to the minimization problem

$$\min_{\boldsymbol{\beta}} \quad -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i\mathbf{x}_i^T\boldsymbol{\beta} - \log(1 + e^{\mathbf{x}_i^T\boldsymbol{\beta}})\Big] \quad \text{subject to} \quad \lambda\sum_{j=1}^{p}|\beta_j| \leq k.$$

The difference is that the lasso uses an ℓ_1 penalty instead of the ℓ_2 penalty used in ridge regression. The ℓ_q-norm of a vector $\boldsymbol{\beta}$ is defined as

$$\|\boldsymbol{\beta}\|_q = \left(\sum |\beta_j|^q\right)^{1/q}.$$

As with ridge regression, the estimates are shrunken towards zero. The only difference is that it is now possible for β_j to actually take the value zero. Figure 3.1 illustrates this situation. The ridge and lasso estimates are given by the first point at which a contour of the negative log-likelihood is in contact with the constraint region from the induced penalty. Since ridge has a circular constraint, this intersection will generally not occur on an axis, causing non-zero estimates. Contrarily, the lasso has corners at each of the axes, and one can therefore often observe estimates equal to zero.

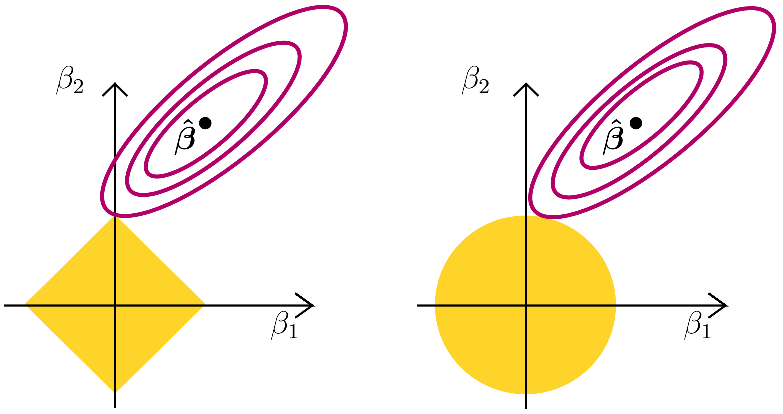


Figure 3.1: Contours of the negative log-likelihood and constraint regions for the lasso (left) and ridge regression (right). The yellow areas are the constraint regions, $|\beta_1| + |\beta_2| \leq k$ and $\beta_1^2 + \beta_2^2 \leq k$, while the pink ellipses are the contours of the negative log-likelihood. The figure is as in [7].

Consequently, models generated by the lasso are much easier to interpret. Nevertheless, when the aim is prediction one must investigate which of the methods produces the best result. Previous studies have shown that none of them uniformly dominates the other [11]. A new regularization technique which combines both methods is therefore proposed as a solution, and is known as the elastic net. Similarly to the lasso, the elastic net performs variable selection and continuous shrinkage simultaneously, as well as selecting groups of correlated variables. Studies have also indicated that the elastic net outperforms the lasso in terms of prediction accuracy. Partly for this reason, Zou and Hastie (2005) [12] introduced the elastic net penalty yielding

$$\begin{aligned}\ell_\lambda^{EN} &= \frac{1}{N}\sum_{i=1}^{N}\left(y_i\mathbf{x}_i^T\boldsymbol{\beta} - \log(1+e^{\mathbf{x}_i^T\boldsymbol{\beta}})\right) - \lambda\sum_{i=1}^{p}\left(\alpha|\beta_j| + \frac{1}{2}(1-\alpha)\beta_j^2\right) \\ &= \frac{1}{N}\ell(\boldsymbol{\beta}) - \lambda\left(\alpha\|\boldsymbol{\beta}\|_1 - \frac{1}{2}(1-\alpha)\|\boldsymbol{\beta}\|_2^2\right),\end{aligned} \tag{3.12}$$

which is a compromise between ridge and lasso. Hence, the elastic net approach selects variables like the lasso, while shrinking coefficients for correlated predictors like ridge. The parameter α determines the weight of the penalties, and must be determined. The most common method in deciding the value of λ and α is cross-validation described in section 3.1.4.

Analogous to the MLEs, $\hat{\boldsymbol{\beta}}^{\lambda}$ using the elastic net is also found using an iterative maximization procedure. If we let $\tilde{\boldsymbol{\beta}}$ denote the current estimates, then the quadratic approximation of the log-likelihood will be

$$\ell_Q = -\frac{1}{2}\sum_{i=1}^{N} w_i(z_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + C(\tilde{\boldsymbol{\beta}}),$$

where C is a constant independent of $\tilde{\boldsymbol{\beta}}$, the weights w_i and working response z_i are

$$\begin{aligned} w_i &= \mathbf{x}_i^T\tilde{\boldsymbol{\beta}} + \frac{y_i - \tilde{\pi}_i}{\tilde{\pi}_i(1-\tilde{\pi}_i)}, \\ z_i &= \tilde{\pi}_i(1-\tilde{\pi}_i), \end{aligned}$$

and $\tilde{\pi}_i$ is evaluated using the current parameter estimates. For each value of λ, we compute the quadratic approximation ℓ_Q about the current parameters $\tilde{\boldsymbol{\beta}}$. Then coordinate descent is used to solve the penalized weighted least squares problem [13]

$$\min_{\boldsymbol{\beta}}\Big\{-\frac{1}{N}\ell_Q(\boldsymbol{\beta}) + \lambda\Big(\alpha\|\boldsymbol{\beta}\|_1 - \frac{1}{2}(1-\alpha)\|\boldsymbol{\beta}\|_2^2\Big)\Big\}. \tag{3.13}$$

This method is partitioned in three loops:

Outer loop: Decrement λ.

Middle loop: Update the quadratic approximation ℓ_Q using the current estimate $\tilde{\boldsymbol{\beta}}$.

Inner loop: Perform the coordinate descent algorithm on problem (3.13).

3.1.4 Model Diagnostics

Two aspects which need to be considered when evaluating a model is the power of predicting, and whether the model is correctly specified. It is important to separate these two. Based on previous data, a model could perform excellent in predicting, but if it does not explain the variability in the data well its predictive power will not be as strong in the future. In this section we will therefore first focus on the goodness of fit, and then the predictive power.

The well known R-squared statistic, which measures the variability in the response explained by a linear regression model, is not applicable for logistic regression as the response is dichotomous rather than continuous. According to [14] there are many ways to calculate a pseudo R^2 for logistic regression, but there is no agreement on which one is the best. The most common approach is the one proposed by McFadden defined as

$$R^2_{\text{McFadden}} = 1 - \ell_C/\ell_0,$$

where ℓ_C and ℓ_0 are the maximum log-likelihoods of the candidate model and null model[1], respectively. The interpretation of this pseudo R-squared measure is discussed in [15], which states that its values tend to be considerably lower than for the standard R^2 measure. McFadden himself actually stated that values between 0.2-0.4 indicate an excellent model fit.

In order to investigate the model fit for a generalized linear model, the deviance given by

$$D = 2\sum_{i=1}^{N} \Big[y_i \log\Big(\frac{y_i}{n_i\hat{\pi}_i}\Big) + (n_i - y_i)\log\Big(\frac{n_i - y_i}{n_i - n_i\hat{\pi}_i}\Big)\Big], \tag{3.14}$$

is often used [16]. We use D when assessing the model fit as it grows large when the model fits poorly. The asymptotic distribution of D, under the assumption that the model is correctly specified, is $D \sim \chi^2_{(N-p)}$. This statistic is also asymptotically equivalent to the Pearson statistic given by

$$X^2 = \sum_{i=1}^{N} \frac{(y_i - n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)} = \sum_{i=1}^{N} e_i^2,$$

which divides the raw residual $(y_i - n_i\hat{\pi}_i)$ by the estimated binomial standard deviation of y_i. The proof of the relationship of D and X^2 is given in [17]. The choice between the two depends on the adequacy of the approximation of the chi-squared distribution, but there is some evidence that X^2 is often the preferred choice, since D is influenced by very small frequencies. However, when explanatory variables are continuous, often the number of distinct groups in a sample is equal to the number of observations, which means that $n_i = 1$ for all i. Additionally, as the response y_i is equal to either 0 or 1, the size of the residuals is limited. Thus, a fatal error in the model will not be recognized, and both D and X^2 may be uninformative.

It is, however, worth noticing that the deviance in (3.14) can also be written as

$$D = -2(\ell_C - \ell_S), \tag{3.15}$$

where ℓ_S is the maximum log-likelihood of the saturated model. The saturated model is defined as the model which provides a perfect fit as it has a separate parameter for each observation. It serves therefore as a baseline for other models, such as exploring model fit. The likelihood-ratio statistic comparing two models is simply the difference between the deviances. Thus, we can use (3.15) in the comparison of two nested models, M_0 and M_1, through the Likelihood Ratio Test

$$\text{LRT} = D_0 - D_1 = -2(\ell_0 - \ell_1),$$

[1] Null model is defined as the model only including the intercept

which is asymptotically chi-squared distributed with the difference in parameters as the number of degrees of freedom. LRT grows large when M_1 is a better fit than M_0. With binomial responses, this test does not depend on whether the data is grouped or not.

The Hosmer-Lemeshow (HL) test is a commonly used procedure for assessing goodness of fit when the deviance and the Pearson statistic no longer are useful. Here, the predicted values are ordered from lowest to highest, and then separated into several groups of approximately equal size. Hence,

$$X^2_{HL} = \sum_{j=1}^{g} \frac{(O_j - E_j)^2}{E_j(1 - E_j/n_j)} \sim \chi^2_{g-2}.$$

For each group g, the observed number of events, O_j, as well as the expected number of events, E_j, are calculated. Additionally, n_j is the number of observations in the jth group. A group size of $g = 10$ is standard, but for $1000 < N < 25000$ a reasonable formula for g is given as [18]

$$g = \max\left(10, \min\left\{\frac{m}{2}, \frac{N-m}{2}, 2 + 8\left(\frac{N}{1000}\right)^2\right\}\right), \tag{3.16}$$

where m is the number of successes. Furthermore, if

$$\mathrm{P}(\chi^2_{g-2} > X^2_{HL,\mathrm{Obs}}) \leq \alpha,$$

we have evidence to reject the null hypothesis stating that the model is correctly specified. It should be emphasized that a large p-value does not necessarily mean the model fits well, since evidence against a null hypothesis is not equivalent to evidence in favour of the alternative hypothesis.

This statistic has become quite popular, but even here we are faced with problems. The most troubling is that the result depends heavily on the number of groups. Furthermore, one would think that adding a statistically significant interaction or non-linear term to a model would improve its fit, judged by the HL test. However, as noted in [14], often this does not happen. It is also experienced that adding a statistically insignificant interaction or non-linear term to a model will improve the HL fit, which is unacceptable behaviour. Actually, it is suggested that when the sample size $N > 25000$, the HL test is not recommended due to the rapid increase of suggested value of g with N in (3.16).

Plots of residuals against the predictors may detect a type of lack of fit, and a common approach is the residuals which uses components of the deviance. In (3.14) let $D = \sum d_i^2$. Then, the deviance residual for observation i is defined as d_i with the condition that its sign is the same as $(y_i - n_i\hat{\pi}_i)$. As mentioned earlier, these plots could turn out to be very uninformative as $n_i = 1$. In this case, it may be necessary to rely on other diagnostics such as Cook's distance, which for a logistic regression model is approximated to be [19]

$$D_i \approx \frac{1}{p}\left(\frac{h_{ii}}{1 - h_{ii}}\right) r_i^2, \tag{3.17}$$

which measures the ith observation's influence by looking at how much the entire regression function changes when removing this observation. The standardized residuals r_i in (3.17) are given as

$$r_i = \frac{e_i}{\sqrt{v_i(1-h_{ii})}}$$

The value h_{ii} is the ith element in the diagonal of the hat matrix

$$\mathbf{H} = \mathbf{V}^{1/2}\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}^{1/2}, \tag{3.18}$$

and is called the leverage of the ith observation. From (3.18) we have that the leverage is a distance measure, which tells us if an observation is located far from the others in the predictor space.The diagonal matrix **V** has the elements $v_i = \hat{\pi}_i(1-\hat{\pi}_i)$, which is the estimated variance of Y_i.

An observation with high leverage can potentially make a substantial difference to the fit. A convention is that if h_{ii} is greater than two or three times p/N, it may be a concern. Large values of (3.17) indicate high influence from observation i. There is, however, no significance test for D_i, but values near or larger than 1 indicate high influence. Values for D_i much larger than the others is sometimes worth investigating.

Another issue in addressing the adequacy of models for binary data is overdispersion. The observations may have a greater variance than $\pi_i(1-\pi_i)$. An indicator of this problem is if the deviance D in equation (3.14) is much greater than $N-p$ (where p is the number of parameters included in the model) [17]. This could be due to for instance important explanatory variables being excluded or if the link function is incorrect.

Probably the simplest and most common used method for estimating prediction error is cross-validation. The approach involves randomly dividing the available dataset into k groups, or folds, of approximately equal size. Each fold is treated as a validation set, and the fit is on the remaining $k-1$ folds. The mean squared error given as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{\text{RSS}}{N},$$

where RSS is the residual sum of squares, is then computed on the observations not included in the model fit. This procedure is repeated k times, which produces k estimates for MSE. The predictive estimate is then found by averaging these values,

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i.$$

The common choice of k is 5 or 10.

Cross-validation is an excellent tool in comparing several classification algorithms. Having obtained, for instance, two learning algorithms we want to compare and test whether their predictive performance are significantly different. To do this we use a k-fold cross-validation, and for each fold i we let the measure of interest be $p_i = p_i^1 - p_i^2$, which is the difference in observed performance. This is a paired t-test where we have a distribution of p_i containing k points. Given that both p_i^1 and p_i^2 are approximately normal, their difference p_i is also normal. The null hypothesis is then that the distribution of p_i has mean zero:

$$H_0 : \mu = 0 \quad \text{vs.} \quad H_1 : \mu \neq 0,$$

where we define

$$\bar{p} = \frac{1}{k}\sum_{i=1}^{k} p_i \quad \text{and} \quad S^2 = \frac{\sum_{i=1}^{k}(p_i - \bar{p})^2}{k-1}.$$

We reject the null hypothesis if

$$T = \frac{\sqrt{k}(\bar{p} - 0)}{S} \sim t_{k-1}$$

lies outside the interval $(-t_{\alpha/2,k-1}, t_{\alpha/2,k-1})$. If the test rejects, then we can conclude that one of the models is significantly better than the other [20].

Other methods of predictive evaluation originates from a confusion matrix which contains information about the actual and predicted values done by a classification model [21].

Table 3.1: Confusion Matrix

		Prediction	
		False	True
Reference	False	True Negative	False Positive
	True	False Negative	True Positive

Using Table 3.1 the accuracy AC of a model is defined as

$$AC = \frac{\text{True Negative} + \text{True Positive}}{\text{Total Observations}},$$

and is the proportion of the total number of predictions that were correct. The precision P is given as the proportion of the predicted positive cases that were correct, calculated by

$$P = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}.$$

Furthermore, the true positive rate TP and false positive rate FP are given as

$$TP = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

and

$$FP = \frac{\text{False Positive}}{\text{True Negative} + \text{False Positive}}$$

respectively. The true positive rate is the proportion of positive cases that were correctly specified, and the false positive rate is consequently the proportion of negative cases that were incorrectly classified.

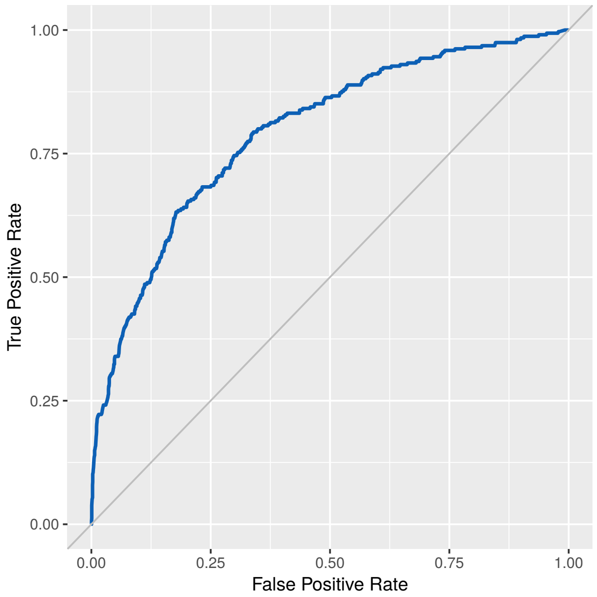


Figure 3.2: Example of a ROC curve

The Receiver Operating Characteristics (ROC) curve is a plot of the TP against the FP, and is used to illustrate the relative tradeoffs between benefits (TP) and costs (FP). For a regression model this is a continuous graph as it contains several confusion matrices based on the threshold of the binary decision. A discrete classifier, however, produces only a single point in the ROC space as it has only one identity matrix. Figure 3.2 presents an example of a ROC curve for a regression model, where the diagonal line, also known as the line of equality, represents the choice of randomly guessing the response. The point $(FP = 0, TP = 1)$ is a perfect classification model as it classifies all positive- and negative cases correctly. The point $(FP = 0, TP = 0)$ is for a model that is predicting all events to be negative, similarly will a model in $(FP = 1, TP = 1)$ predict all events to be positive. Finally, $(FP = 1, TP = 0)$ represents a model which will predict incorrectly for all events. It is therefore desired for a regression model to have a ROC curve close to the top left corner of Figure 3.2. An effective measure to represent a ROC performance is the area under the curve (AUC). It ranges between 0 and 1, but it is worth mentioning that an AUC of 0.5 describes a model of random guessing.

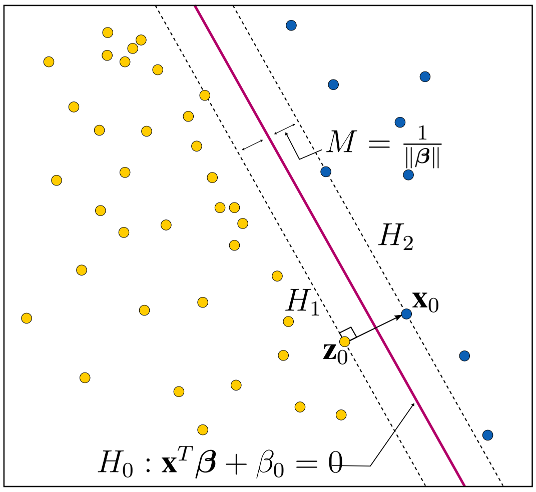
3.2 Support Vector Machine

Support Vector Machine (SVM) is a classification approach developed in the computer science community in the 1900s, and has since then become quite popular. As for logistic regression, we have N data points $(y_i, x_{1i}, ..., x_{ik}), i = 1, ..., N$, but now the categories of y_i is coded -1

and 1. We will first assume that the two classes are linearly separable, meaning there exist a hyperplane given as

$$\{H_0 : \mathbf{x} : f(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta} + \beta_0 = 0\}, \tag{3.19}$$

which separates the two classes entirely illustrated by Figure 3.3. Here, $\boldsymbol{\beta}$ is normal to the hyperplane.



$M = \frac{1}{\|\boldsymbol{\beta}\|}$
H_2
$\mathbf{x}_0$
H_1
$\mathbf{z}_0$
$H_0 : \mathbf{x}^T\boldsymbol{\beta} + \beta_0 = 0$

Figure 3.3: Illustration of the separable case. The decision boundary is the solid line, while the dashed lines are the maximal margin of width $2M = 2/\|\boldsymbol{\beta}\|$.

The data can then be described by

$$\begin{aligned}
&H_1: \ f(\mathbf{x}_i) \geq 1 \quad &&\text{for} \quad y_i = 1\\
&H_2: \ f(\mathbf{x}_i) \leq -1 \quad &&\text{for} \quad y_i = -1\\
&\Leftrightarrow \quad y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - 1 \geq 0, \quad &&\forall \mathbf{x}_i, y_i.
\end{aligned}$$

In other words, an observation is assigned to either class depending on which side of the hyperplanes, H_1 and H_2, it is located. If such planes exist, there exist an infinite number of such hyperplanes. This is because a given hyperplane can be shifted or rotated a tiny amount and still separate the classes perfectly. If we have an observation $\mathbf{x}_0$ which satisfies

$$y_i(\mathbf{x}_0^T\boldsymbol{\beta} + \beta_0) - 1 = 0, \tag{3.20}$$

then this point is known as a support vector, lying exactly on either of the boundaries between the two classes. Our aim is now to find which values of β_0 and $\boldsymbol{\beta}$ that produce the biggest distance between H_1 and H_2. If we let $\mathbf{x}_0$ lie on H_2 and $\mathbf{z}_0$ on H_1 such that $(\mathbf{z}_0 - \mathbf{x}_0)\perp H_2$, then since $\boldsymbol{\beta}$ is orthogonal to H_1 and H_2 we can write

$$\mathbf{z}_0 = \mathbf{x}_0 + (\mathbf{z}_0 - \mathbf{x}_0) = \mathbf{x}_0 + 2M\frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|}, \tag{3.21}$$

where $2M$ is the distance between the two hyperplanes. We have also that

$$\mathbf{z}_0^T\boldsymbol{\beta} + \beta_0 = 1 \quad \text{and} \quad \mathbf{x}_0^T\boldsymbol{\beta} + \beta_0 = -1. \tag{3.22}$$

Inserting (3.22) into (3.21)

$$\begin{aligned} 1 &= (\mathbf{x}_0 + 2M\frac{\boldsymbol{\beta}}{||\boldsymbol{\beta}||})^T\boldsymbol{\beta} + \beta_0 \\ &= \mathbf{x}_0^T\boldsymbol{\beta} + 2M||\boldsymbol{\beta}|| + \beta_0 \\ &= -1 + 2M||\boldsymbol{\beta}|| \\ \Rightarrow \quad M &= \frac{1}{||\boldsymbol{\beta}||}. \end{aligned}$$

Hence, maximizing M is equivalent to minimizing $||\boldsymbol{\beta}||$. Since minimizing $||\boldsymbol{\beta}||$ is equivalent to minimizing $\frac{1}{2}||\boldsymbol{\beta}||^2$, our problem can be stated as [22]

$$\begin{aligned} \min \quad & \frac{1}{2}||\boldsymbol{\beta}||^2 \\ \text{subject to} \quad & y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - 1 \geq 0, \quad i = 1, ..., N, \end{aligned} \tag{3.23}$$

and makes it possible to perform Quadratic Programming (QP) optimization later.

Suppose now that there does not exist a hyperplane which is able to separate the two classes perfectly. In other words, we have an overlap of observations. One way to deal with this overlap is to allow for some observations to be misclassified. We still aim to minimize $||\boldsymbol{\beta}||$, but now we allow for some slack denoted $\xi = (\xi_1, ..., \xi_N)$. Hence, we obtain the optimization problem [23]

$$\begin{aligned} \min_{\xi} \quad & \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) \geq 1 - \xi_i, \quad \forall i \\ & \xi_i \geq 0, \end{aligned} \tag{3.24}$$

where $C \geq 0$ denotes the cost parameter controlling the penalty paid for the misclassification of an observation. If $\xi_i = 0$ then the ith observation is on the correct side of the margin. If $\xi_i > 0$ then the ith observation is on the wrong side of the margin, and if $\xi_i > 1$ then it is on the wrong side of the hyperplane. Hence, ξ_i measures the degree of misclassification of x_i. In (3.24), C bounds the sum of the ξ_i's. If $C = 0$ then there is no acceptance for violations to the margin. For $C > 0$ no more than C observations can be on the wrong side of the hyperplane. In other words, increasing C makes our model more tolerant of misclassification, and its value is often chosen using cross-validation. One interesting property of this problem statement is that only observations which lie on or on the wrong side of the margin affect the hyperplane, and is the reason to why these observations are known as the support vectors.

We can describe the solution of (3.24) using Lagrange multipliers. The Lagrange primal function is

$$L_P = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1-\xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i, \tag{3.25}$$

which we minimize with respect to $\beta_0, \boldsymbol{\beta}$ and ξ_i. Setting the respective derivatives of (3.25) to zero, we obtain

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i \tag{3.26}$$

$$\frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow 0 = \sum_{i=1}^{N}\alpha_i y_i \tag{3.27}$$

$$\frac{\partial L_P}{\partial \xi_i} = 0 \Rightarrow \alpha_i = C - \mu_i, \quad \forall i, \tag{3.28}$$

in addition to the positivity constraints $\alpha_i, \mu_i, \xi_i \geq 0 \forall i$. Substituting (3.26)-(3.28) into (3.25) we get the Lagrangian dual function

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j,$$

which is maximized with respect to $\alpha_i \in [0, C]$ and (3.27), and yields a lower bound of (3.24) for any feasible point. The Karush-Kuhn-Tucker conditions include the constraints

$$\alpha_i[y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1-\xi_i)] = 0 \tag{3.29}$$

$$\mu_i\xi_i = 0 \tag{3.30}$$

$$y_i(\mathbf{x}_i^T\boldsymbol{\beta} + \beta_0) - (1-\xi_i) = 0, \tag{3.31}$$

$\forall i$. Together the equations (3.26)-(3.31) will uniquely characterize the solution to both the primal and dual problem. From (3.26) we have the solution for $\boldsymbol{\beta}$

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N}\hat{\alpha}_i y_i \mathbf{x}_i, \tag{3.32}$$

and since any observation satisfying (3.20) is a support vector $\mathbf{x}_s$, it will have the form

$$y_s(\mathbf{x}_s^T\boldsymbol{\beta} + \beta_0) = 1.$$

We can now find β_0 by solving

$$y_s(\mathbf{x}_s^T \sum_{m \in S} (\hat{\alpha}_m y_m \mathbf{x}_m) + \beta_0) = 1,$$

for β_0, where S denotes the set of support vectors. This method is known as the support vector classifier, and is how we deal with non-separable observations. But what if the boundary between the two classes is not linear? In order to address this problem, we introduce the concept of support vector machine.

As with other linear methods, we can make the procedure more flexible by enlarging the feature space using polynomials or splines. The support vector machine is an extension of this idea, where the dimension of the feature space is enlarged in a specific way using the concept of kernels. We notice from (3.32) that the solution of the support vector classifier problem simply involves the inner product of the observations, and not the observations themselves. The inner product between two observations $\mathbf{x}_i$ and $\mathbf{x}_j$ is given as

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{m=1}^{k} x_{im} x_{jm}.$$

The Lagrange dual function can therefore by re-expressed as

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle, \tag{3.33}$$

where $\Phi(\mathbf{x})$ is a transformed feature vector. The solution function given in (3.19) can therefore be written

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle + \beta_0. \tag{3.34}$$

For both (3.33) and (3.34) is $\Phi(\mathbf{x})$ involved only through inner products. Hence, it is not necessary to specify $\Phi(\mathbf{x})$, but rather the kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle, \tag{3.35}$$

which computes the inner product of the transformed space, quantifying the similarity between two observations. It demands far less computational effort than explicitly projecting $\mathbf{x}_i$ and $\mathbf{x}_j$ into the feature space of $\Phi(\mathbf{x})$. This allows for the transformation of a non-linear problem into a higher dimensional linear problem which produces more accurate predictions.

When there is no prior knowledge regarding the data the Gaussian- and Laplace RBF kernels are popular choices, and are given as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{x}_j\|),$$

respectively.

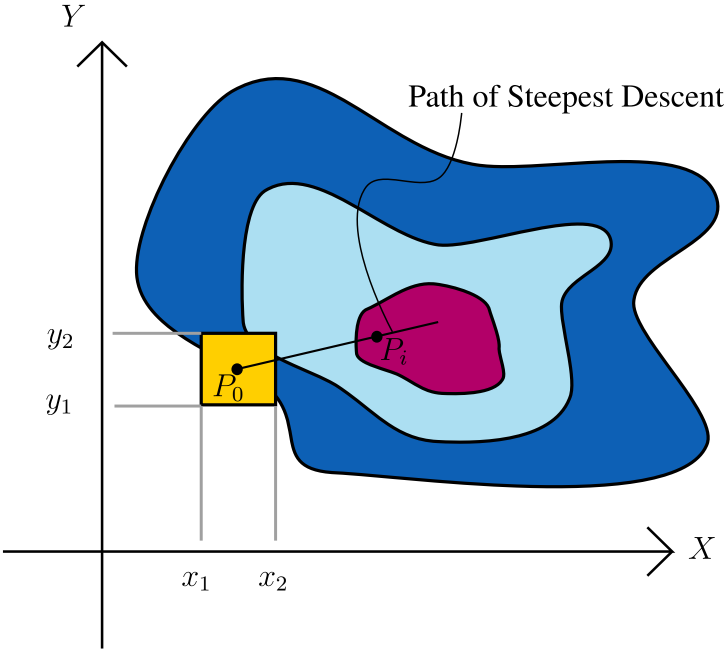


Figure 3.4: An illustration of the path of steepest descent for a two-factor experiment

When introducing the concept of kernels, the issue regarding the choice of the hyper parameters arises. In order to decide the hyper parameter values, here C and σ, which produces the model with minimum predictive error, the concept of response surface methodology (RSM) is introduced [24]. The reason for applying RSM as an optimization tool is due to its cost efficient performance. Instead of iteratively computing the predictive error for multiple combinations of the tuning parameters, a selection of points inside the operating conditions is chosen, making up a two level factorial experiment. It is now necessary to explore the chosen region to decide which direction needs to be taken to move towards the optimum region. Hence, an estimation of the path of steepest descent is made. The method of steepest descent is an algorithm for finding the nearest local minimum of a function which presupposes that the gradient of the function can be computed. This method starts at a point P_0 and, as many times as needed, moves from P_i to P_{i+1} by minimizing along the line extended from P_i in the direction of $-\nabla f(P_i)$, the local downhill gradient. Looking at Figure 3.4 the initial point P_0 is chosen to be the centre of the $2^k = 2^2$ level experiment with the levels $\{x_1, x_2\}$ and $\{y_1, y_2\}$ resulting in a cuboidal experiment region made up of four factorial points. The purpose is to find the parameter values where the response is at its minimum, which in Figure 3.4 is given as the purple region. If some notion about where the optimum should be is present, then this is a good area to start. The idea is to keep experimenting along the direction of steepest descent until there is no further improvement on the predictive error.

Normally, one follows the gradient until stabilization, and then perform another 2^k experiment with a new centre point. One alternative here is to modify the gradient, or expand the experiment by adding a group of axial points around the evaluated point P_i resulting in a central composite design (CCD) illustrated by Figure 3.5. The CCD is also the most commonly used design to estimate a second-order model as each factor here has five different levels. If the factorial points are denoted ± 1, then the axial points are given as $\pm\alpha$ ($\alpha > 1$). To preserve rotatability, defined as a design which can be rotated around its centre without changing the prediction variance, the value of α depends on the number of factors involved through [25]

$$\alpha = (2^k)^{1/4}.$$

Even though this is the standard way to do this kind of optimization, other approaches are possible depending on what is observed.

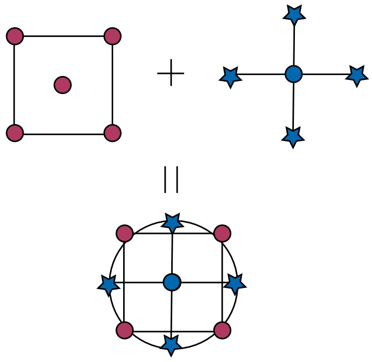


Figure 3.5: Diagram of central composite design generation for two factors

3.3 Fuzzy Clustering

3.3.1 Fuzzy Logic

Cluster analysis belongs to the unsupervised way of statistical modelling, where no prior information regarding the data is needed. The aim is to obtain useful information regarding the structure of a given dataset by partitioning the observations into distinct groups. These groups are built such that the observations within each group resemble each other, while observations between groups differ from each other. Clustering is popular in many fields, resulting in a great number of methods. One of the most common approaches is known as K-means clustering, where N observations consisting of p measured variables are assigned to a predefined number of non-overlapping clusters. This procedure results from a simple and intuitive mathematical problem, where we begin by defining $C_1, ..., C_K$ as the sets containing the indices of the observations in each cluster. The set of observations is represented as an $N \times p$ matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{Np} \end{bmatrix}, \tag{3.36}$$

denoted as the pattern or data matrix. Here, the rows of the matrix are called patterns or objects, while the columns are called the features or attributes.

The sets $\{C_k : k = 1, ..., K\}$ must satisfy the following properties [26]:

$$\bigcup_{k=1}^{K} C_k = \mathbf{X} \tag{3.37a}$$

$$C_k \cap C_{k'} = \varnothing \quad \forall \quad k \neq k' \tag{3.37b}$$

$$\varnothing \subset C_k \subset \mathbf{X} \quad \forall \quad k. \tag{3.37c}$$

Equation (3.37a) states that the union of subsets C_k contains all the observations, while (3.37b) indicate that these subsets are non-overlapping. None of them are empty nor include all the observations as given in (3.37c). Traditionally, observations are judged as to whether they belong to a given set or not. However, in the case of fuzzy sets, whether an observation belongs to a set or not is unclear. In order to represent this mathematically, we utilize the degree of belongingness of each observation to a set. A fuzzy set C_k is a set characterized by the membership function given as

$$\mu_k : \mathbf{X} \to [0, 1], \quad \forall k. \tag{3.38}$$

In other words, $\mu_k(\mathbf{x})$ is the degree of belongingness of $\mathbf{x}$ to a fuzzy set C_k. If the grade $\mu_k(\mathbf{x})$ is close to 1, then our confidence in $\mathbf{x}$ belonging to C_k is great, while a value of $\mu_k(\mathbf{x})$ close to 0 indicates that it is unlikely that $\mathbf{x}$ belongs to C_k.

3.3.2 Fuzzy *c*-Means Clustering

Fuzzy sets were first proposed as a method of capturing the uncertainty present in real data. This soft way of clustering allows for observations to belong to several clusters simultaneously. The hard approach to clustering may fail as its rigid nature is not able to handle real-life complexity. Suppose we have our dataset $\mathbf{X}$ as given in (3.36), and a set of fuzzy clusters which are defined as in (3.38). Then the degree of belongingness of an observation i to a cluster k is denoted $u_{ik} \equiv \mu_k(\mathbf{x}_i)$, and satisfies the conditions

$$u_{ik} \in [0, 1], \forall i, k \quad \text{and} \quad \sum_{k=1}^{K} u_{ik} = 1, \forall i. \tag{3.39}$$

We let $\mathbf{U} = (u_{ik})_{N\times K}$ be the fuzzy partition matrix containing membership values of the ith observation to the kth cluster C_k. An example of a fuzzy partition matrix for a dataset $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_5)^T$ which is to be divided into $K = 2$ clusters, is

$$\mathbf{U} = \begin{bmatrix} 1.0 & 0.0 \\ 0.8 & 0.2 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \\ 0.1 & 0.9 \end{bmatrix}.$$

For this partition matrix, the object $\mathbf{x}_3$ has an equal membership degree in both clusters.

The clusters C_k are described by their member observations and their centre. The centre is often called the prototype, and can be viewed as the most central member of a cluster. Since they are not known beforehand they are sought for simultaneously as the partitioning of the data. Usually, centroids are used as the centres of the clusters, and are the points to which the sum of distances from all objects in that cluster is minimized. We therefore need a partitioning clustering algorithm, which divides $\mathbf{X}$ into K clusters with a low within-cluster variation, and high between-cluster variation. In other words, we seek the partitioning which connects similar observations and distinguish different observations.

Let d_{ij} denote the distance in $\mathbb{R}^p$ between two observations $\mathbf{x}_i$ and $\mathbf{x}_j$; each $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})$. We assume that for all $\mathbf{x}_i$ in $\mathbb{R}^p$ this function satisfies [26]

$$d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{3.40a}$$

$$d_{ij} = 0 \Leftrightarrow \mathbf{x}_i = \mathbf{x}_j \tag{3.40b}$$

$$d_{ij} = d_{ji} \tag{3.40c}$$

Functions which satisfies (3.40) are known as measures of dissimilarity, which means that we can construct a similarity measure s from d. The idea is to incorporate the distances $\{d_{ij}\}$ as a clustering criteria. Fuzzy c-means (FCM) is the method which minimizes the weighted within-class sum of squares [27]

$$J_{FCM}(\mathbf{U}, \mathbf{X}, \mathbf{A}, \mathbf{v}) = \sum_{n=1}^{N} \sum_{k=1}^{K} (u_{nk})^m d^2_{\mathbf{A}}(\mathbf{x}_n, \mathbf{v}_k), \tag{3.41}$$

where $\mathbf{v}_k$ denotes the centroid of the kth cluster, and $d^2_{\mathbf{A}}(\mathbf{x}_n, \mathbf{v}_k)$ is the distance measure between $\mathbf{x}_n$ and $\mathbf{v}_k$ given as

$$d^2_{\mathbf{A}}(\mathbf{x}_n, \mathbf{v}_k) = \|\mathbf{x}_n - \mathbf{v}_k\|^2_{\mathbf{A}} = (\mathbf{x}_n - \mathbf{v}_k)^T \mathbf{A} (\mathbf{x}_n - \mathbf{v}_k) = d^2_{\mathbf{A}nk}.$$

The incorporation of a matrix $\mathbf{A}$, a $(p \times p)$ positive definite matrix, results in a distance weighting according to the statistical properties of the features [28].

The value of m is chosen beforehand from $[0, \infty)$, and represents the degree of fuzziness of the clustering. The goal is to obtain values for the partitions **U** and cluster prototypes $\mathbf{v}_k$. This problem can be solved using multiple methods, including simulated annealing, iterative minimization or genetic algorithms. The most popular method is, however, a simple Picard iteration through the first-order conditions for stationary points of (3.41).

These stationary points can be found by adjoining constraint (3.39) to J_{FCM} by the means of Lagrange multipliers

$$\bar{J}_{FCM}(\mathbf{U}, \mathbf{X}, \mathbf{v}, \boldsymbol{\lambda}) = \sum_{k=1}^{K}\sum_{n=1}^{N}(u_{nk})^m d_{\mathbf{A}nk}^2 + \sum_{n=1}^{N}\lambda_k\Big[\sum_{k=1}^{K}u_{kn} - 1\Big],$$

and letting the gradients of $\bar{J}_{FCM}$ with respect to **U**, **V** = $(\mathbf{v}_1, ..., \mathbf{v}_c)^T$ and $\boldsymbol{\lambda}$ be equal to zero. It can be shown that requiring $m > 1$ may result in the minimization of (3.41) only if

$$u_{nk} = \frac{1}{\sum_{j=1}^{K}(d_{\mathbf{A}nk}/d_{\mathbf{A}nj})^{2/(m-1)}}, \quad \forall k, n, \tag{3.42}$$

and

$$\mathbf{v}_k = \frac{\sum_{n=1}^{N}(u_{nk})^m \mathbf{x}_n}{\sum_{n=1}^{N}(u_{nk})^m}, \quad \forall k, \tag{3.43}$$

which are both first-order necessary conditions for stationary points of (3.41). The FCM presented in Algorithm 1 iterates through (3.42) and (3.43). Its convergence, and the sufficiency of (3.42) and (3.43) is proven in [26].

It is worth noticing a few remarks regarding the FCM algorithm. First of all, this algorithm finds a local optimum instead of a global, which results in the final partitioning being dependent on the initial cluster assignments $\mathbf{U}^{(0)}$. Furthermore, while step **1.** and **2.** in Algorithm 1 are quite straight forward, step **3.** is a bit more complex, since a singularity will occur when $d_{\mathbf{A}nk}^2 = 0$ for some $\mathbf{x}_n$ and one or more $\mathbf{v}_k$. This rarely happens, but results in zero memberships assigned to the cluster for which $d_{\mathbf{A}nk}^2 > 0$. For the remaining clusters the memberships will be distributed arbitrarily. The *if otherwise* statement at this step is built to handle this singularity. Also, when presenting Algorithm 1 we take for granted that the values of $K, m, \mathbf{U}^{(0)}$, **A** and ε is known when this, in fact, is not the case. It is therefore important to run the algorithm multiple times with different initial values, and then choose the one producing the lowest value of J_{FCM}.

The number of clusters K is by far the most important parameter in the sense that the other parameters have far less influence on the resulting partition. When dealing with real data without any prior information on the data structure, one usually has to make some assumptions. The FCM algorithm will then search for K clusters regardless of whether they are present or not. There are mainly two approaches in determining the appropriate number number of clusters: validity measures and iterative merging.

Algorithm 1 Fuzzy c-Means (FCM)

```
    Given a dataset $\mathbf{X}$, choose the number of clusters $1 < K < N$, the degree of fuzziness
    $m > 1$, the norm-inducing matrix $\mathbf{A}$ and the termination tolerance $\varepsilon > 0$. Initialize the
    partition matrix $\mathbf{U}$ randomly.
    Repeat for $i = 1, 2, ...$
 1. Compute the cluster prototypes for all the clusters:
        $$\mathbf{v}_k^{(i)} = \frac{\sum_{n=1}^{N} (u_{nk}^{(i-1)})^m \mathbf{x}_n}{\sum_{n=1}^{N} (u_{nk}^{(i-1)})^m}, \quad \forall k$$
 2. Compute the distances:
        $$d^2_{\mathbf{A}nk} = \|\mathbf{x}_n - \mathbf{v}_k^{(i)}\|_{\mathbf{A}}, \quad \forall k, n$$
 3. Update the degree of membership of all feature vectors in all the clusters:
    If $d_{\mathbf{A}nk} > 0 \quad \forall i$
        $$u_{nk}^{(i)} = \frac{1}{\sum_{j=1}^{K} (d_{\mathbf{A}nk}/d_{\mathbf{A}nj})^{2/(m-1)}},$$
    otherwise
        $$u_{nk}^{(i)} = 0 \quad \text{if} \quad d_{\mathbf{A}nk} > 0, \quad \text{and} \quad u_{nk}^{(i)} \in [0, 1] \quad \text{with} \quad \sum_{k=1}^{K} u_{nk}^{(i)} = 1$$
    Until $\|\mathbf{U}^{(i)} - \mathbf{U}^{(i-1)}\| < \varepsilon$
```

The former are scalar indices that assess the goodness of the obtained partition. If the value of K equals the number of groups actually present in the data, it is expected that the algorithm will identify them correctly. When this is not the case, misclassifications will appear and the clusters will consequently not be well separated nor compact. Validity measures are therefore built to quantify the separation and compactness of clusters, and are presented later in section 3.3.5.

The latter starts with a sufficiently large number of clusters, and successively reduce this number by merging similar clusters with respect to some predefined criteria [29]. The opposite, which starts with a small number of clusters and iteratively split clusters where the observations have a low degree of membership, is also possible.

The fuzziness parameter m is also quite important as it significantly influences how soft the partition is going to be. Saving the reasoning for section 3.3.5, an interval of $[1.5, 2.5]$ for the value of m is recommended [30].

It is standard practice to set the initial prototypes uniformly at random from $\mathbf{X}$, but the approach of augmenting the c-means algorithm with a simple, randomized seeding technique, known as k-means++, has been shown to improve both the speed and the accuracy dramatically [31]. This specific way of choosing initial prototypes for the c-means algorithm is to first define $D(\mathbf{x})$ as the shortest distance from a data point to the closest centre that has already been chosen.

Then, the algorithm runs as follows:

1. Take the first centre $\mathbf{v}_1$ chosen randomly from $\mathbf{X}$.
2. Take a new centre $\mathbf{v}_i$, choosing $\mathbf{x} \in \mathbf{X}$ with the probability $\frac{D(\mathbf{x})^2}{\sum_{\mathbf{x}\in\mathbf{X}} D(\mathbf{x})^2}$.
3. Repeat step 2. until we have K centres altogether.

Another method, proposed by Al-Daoud [32], performs its calculation the following way

1. Compute the variance of each feature.
2. Detect the feature with the highest variance, denote it $cvmax$, and sort it in any order.
3. Divide the points of $cvmax$ into K subsets of equal length.
4. Find the median of each subset, and use the corresponding vectors for each median to initialize the cluster prototypes.

This initialization will presumably only be effective for datasets where the variability is concentrated in one dimension, as it only considers the feature with highest variance.

The shape of the clusters is determined by the norm-inducing matrix $\mathbf{A}$. A common choice is $\mathbf{A} = \mathbf{I}$, which is the standard Euclidean norm, and is used if it is believed that correlations between the features are low. This might, however, not be the case as illustrated in Figure 3.6, and we therefore need a method which tolerates clusters to be of different shapes.

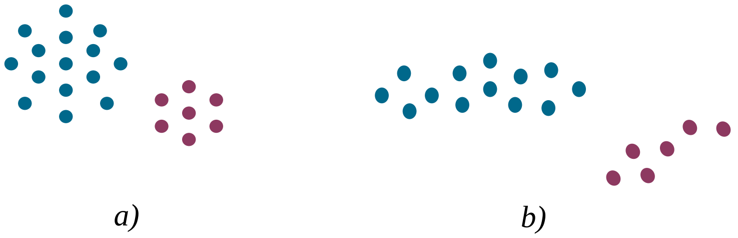


Figure 3.6: Clusters of different shapes and dimensions in $\mathbb{R}^2$.

Finally, a termination parameter value of $\varepsilon = 0.001$ is the most common choice. However, a value of $\varepsilon = 0.01$ is often chosen as it works well in practice, while drastically reducing the computational effort.

3.3.3 Gustafson-Kessel Algorithm

Gustafson and Kessel [28] introduced an adaptive distance norm, which served as a detection of clusters of different geometrical shapes in a dataset. Each cluster would have its own norm-inducing matrix $\mathbf{A}_k$, yielding the objective functional of the GK algorithm:

$$J_{GK}(\mathbf{U}, \mathbf{X}, \{\mathbf{A}_k\}, \mathbf{v}) = \sum_{n=1}^{N} \sum_{k=1}^{K} (u_{nk})^m d^2_{\mathbf{A}_k}(\mathbf{x}_n, \mathbf{v}_k),$$

where $d_{\mathbf{A}_k}$ is the Mahalanobis distance. Since the objective function is linear in $\mathbf{A}_k$, it cannot be directly minimized with respect to $\mathbf{A}_k$. Hence, a feasible solution is only obtained if $\mathbf{A}_k$ contains some restrictions. The most common choice is to constrain the determinant of $\mathbf{A}_k$ such that

$$|\mathbf{A}_k| = \rho_k, \quad \rho_k > 0, \quad \forall k.$$

Fixing the determinant corresponds to optimizing the cluster's shape while keeping the volume constant. Using the Lagrange-multiplier method yields an expression for $\mathbf{A}_k$

$$\mathbf{A}_k = [\rho_k \det(\mathbf{F}_k)]^{1/N} \mathbf{F}_k^{-1},$$

where $\mathbf{F}_k$ is the fuzzy covariance matrix of the kth cluster, and is given as

$$\mathbf{F}_k = \frac{\sum_{n=1}^{N} (u_{nk})^m (\mathbf{x}_n - \mathbf{v}_k)(\mathbf{x}_n - \mathbf{v}_k)^T}{\sum_{n=1}^{N} (u_{nk})^m}.$$

The GK algorithm is presented in Algorithm 2. We note that matrix $\mathbf{A}_k$ is adapted automatically, and does not need to be specified. However, the cluster volumes ρ_k must now be determined. When no prior knowledge is attainable, ρ_k is simply fixed at 1 for each cluster, which causes the drawback of the GK algorithm only being able to detect clusters of similar volumes.

One of the factors influencing the partitioning of clusters is the distance measure. Due to recent advances in fuzzy clustering it is now possible to detect, not only hypervolume clusters, but also clusters shaped as curves and surfaces [33]. Both the FCM and GK include the probabilistic constraint in (3.39) stating that all memberships of a data point across all clusters must sum to 1, and is used in order to avoid the trivial solution of no observation belonging to any cluster. However, since the memberships generated by this constraint are relative numbers, they are not suitable for applications where the aim of the memberships is to represent typicality/compatibility with an elastic constraint. The following example illustrates this problem. Figure 3.7 represents a situation containing two clusters. The FCM/GK would produce very different membership values for the points A and B even though they are equidistant to the prototype, thus equally typical for that particular cluster. This is due to restriction (3.39), which causes point B to transfer some of its connection from cluster 1 to cluster 2. Furthermore, point A and C might obtain equal membership values in cluster 1 although point C is more typical than point A. In other words, the membership values for the FCM and GK algorithm are just relative numbers depending on the membership values for all the other observations.

FCM and many of its derivatives have been proven to be very successful on many clustering problems. However, even here are we faced with major drawbacks as these methods have problems with high dimensional datasets and large number of prototypes [34]. For instance, the fuzzifier function for FCM is an exponential function u^m with $m > 1$. Figure 3.9 visualize the impact of dimensionality to the FCM, where the prototypes move straight into the centre of

Algorithm 2 Gustafson-Kessel (GK) algorithm

```
    Given a dataset X, choose the number of clusters $1 < c < N$, the degree of fuzziness
    $m > 1$, the cluster volumes $\rho_k$ and the termination tolerance $\varepsilon > 0$. Initialize the partition
    matrix U randomly.
    Repeat for $i = 1, 2, ...$
 1. Compute the cluster prototypes for all the clusters:
        $$\mathbf{v}_k^{(i)} = \frac{\sum_{n=1}^{N} (u_{nk}^{(i-1)})^m \mathbf{x}_n}{\sum_{n=1}^{N} (u_{nk}^{(i-1)})^m}, \quad \forall k$$
 2. Compute the fuzzy cluster covariance matrices:
        $$\mathbf{F}_k = \frac{\sum_{n=1}^{N} (u_{nk}^{(i-1)})^m (\mathbf{x}_n - \mathbf{v}_k^{(i)})^T (\mathbf{x}_n - \mathbf{v}_k^{(i)})}{\sum_{n=1}^{N} (u_{nk}^{(i-1)})^m}, \quad \forall k$$
 3. Compute the distances:
        $$d^2_{\mathbf{A}_k nk} = (\mathbf{x}_n - \mathbf{v}_k^{(i)}) \Big[ \rho_k \det(\mathbf{F}_k)^{1/N} \mathbf{F}_k^{-1} \Big] (\mathbf{x}_n - \mathbf{v}_k^{(i)})^T, \quad \forall k, n$$
 4. Update the degree of membership of all feature vectors in all the clusters:
    If $d_{\mathbf{A}_k nk} > 0 \quad \forall i$
        $$u_{nk}^{(i)} = \frac{1}{\sum_{j=1}^{K} (d_{\mathbf{A}_k nk} / d_{\mathbf{A}_k nj})^{2/(m-1)}},$$
    otherwise
        $$u_{kn}^{(i)} = 0 \quad \text{if} \quad d_{\mathbf{A}_k nk} > 0, \quad \text{and} \quad u_{nk}^{(i)} \in [0, 1] \quad \text{with} \quad \sum_{k=1}^{K} u_{nk}^{(i)} = 1$$
    Until $\|\mathbf{U}^{(i)} - \mathbf{U}^{(i-1)}\| < \varepsilon$
```

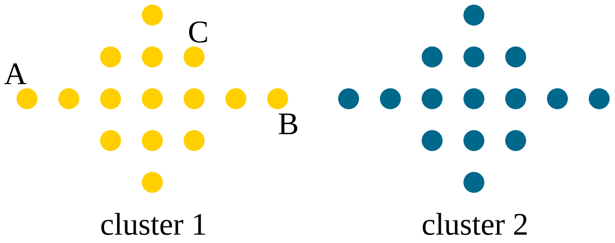


Figure 3.7: Example of a dataset with two clusters where the membership values constructed by the FCM/GK are different for A and B, despite them being equally typical. A and C will have similar membership values regardless of not being equally typical.

gravity of the high dimensional data independent of their initialization, creating no clusters at all. In order to gain some knowledge to why the behaviour presented in Figure 3.9 occurs, Winkler et al. (2012) [34] tested the FCM in a rather artificial way by letting $\alpha \in [0, 1]$ control the location of the prototypes: Let $D = (x_1, ..., x_K) \subset \mathbb{R}^p$ be a dataset containing $K > p$ clusters, with one data object per cluster. The clusters in D are then located on a p-dimensional hypersphere surface arranged such that the minimal pairwise distance is maximized. D is considered as a perfect dataset for clustering due to its clusters being infinitely dense and maximally separated. There is only one small limitation to this statement, which is that K cannot be extremely larger than p (i.e $K < p!$) since the hypersphere surface might be too small to handle an extreme amount of prototypes. Hence, algorithms with problems on D will have even more problems on other datasets as there does not exist a more manageable dataset than D. If $x_i \in \mathbb{R}^p$ is the ith data object with $\mu_D^* \in \mathbb{R}^p$ as the centre of gravity of D, then $v_i(\alpha) = \alpha x_i + (1 - \alpha)\mu_D^*$ and $d_{ij}(\alpha) = d(v_i(\alpha), x_j)$. As the objective function J_{FCM} is a function of the membership values (which are functions of the distance values) and the distance values, it is possible to plot it as a function of α as illustrated in Figure 3.8.

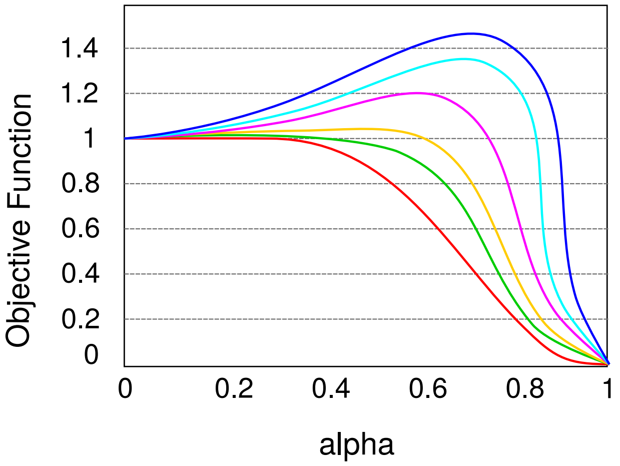


Figure 3.8: Illustration of normalized objective function plots for FCM as a function of α from [34]

This plot presents a strong local maximum between $\alpha = 0.5$ and $\alpha = 0.9$, and studies made by Winkler et al. (2011) [35] showed that the number of dimensions in the dataset effects the objective function by the height of this local maximum. The prototypes however influences the location of the maximum, as a higher number of prototypes moved the local maximum towards higher values of α. Since FCM is a gradient descent algorithm, the prototypes will move to the centre of gravity if their initialization corresponds to a value of α lower than the value producing the local maximum, which is exactly what happens on the right hand side of Figure 3.9. For a p-dimensional hypersphere it is almost impossible to initialize a prototype near enough to a cluster so that it converges to that cluster, since the volume increases exponentially with its radius.

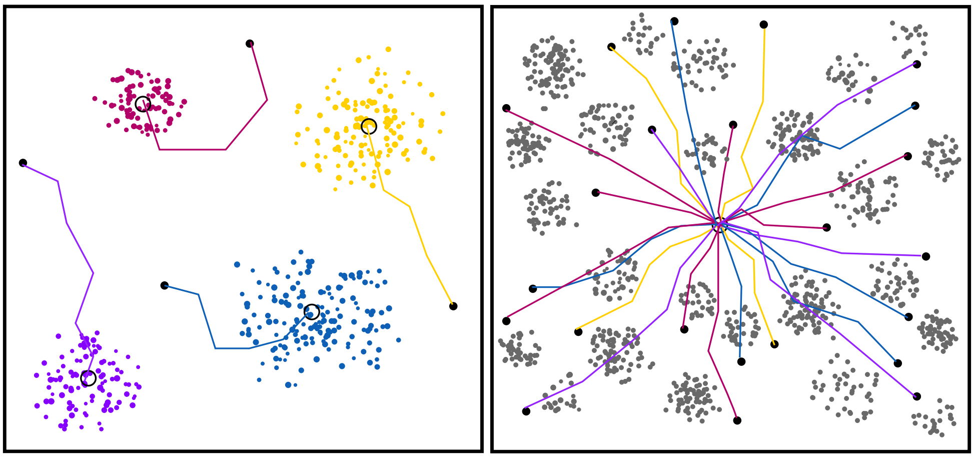


Figure 3.9: Illustration of FCM applied on data with both a low (left) and high (right) number of clusters. The prototypes are portrayed as black circles with tails representing the way the prototypes took from their initialization to their final location from [34].

3.3.4 Possibilistic Fuzzy *c*-means

In order to avoid the problem of dimensionality, Pal et al. [36] proposed a new method called possibilistic fuzzy c-means (PFCM) which produces memberships and possibilities simultaneously. The PFCM minimizes the functional

$$J_{PFCM}(\mathbf{U}, \mathbf{X}, \mathbf{v}, \mathbf{T}) = \sum_{n=1}^{N} \sum_{k=1}^{K} (a u_{nk}^{m} + b t_{nk}^{\eta}) d_{\mathbf{A}_k}^2 (\mathbf{x}_n, \mathbf{v}_k) + \sum_{k=1}^{K} \gamma_k \sum_{n=1}^{N} (1 - t_{nk})^{\eta}, \tag{3.44}$$

with $a > 0, b > 0, m > 1$ and $\eta > 1$. We introduce $\mathbf{T} = [t_{nk}]_{N \times K}$, where each element satisfies

$$t_{nk} \in [0, 1], \quad \forall k, n, \tag{3.45}$$

to represent the typicality matrix, and $\gamma_k > 0$ as a user defined constant. Note that t_{nk} does not have the probabilistic restriction which u_{nk} has. The first term in (3.44) is familiar from (3.41) which demands the distances within a cluster to be as low as possible, while the second term forces t_{nk} to be as large as possible in order to avoid the trivial solution $t_{nk} = 0 \ \forall k, n$. The constraints are now defined as

$$\begin{aligned}\sum_{k=1}^{K} u_{nk} = 1, \quad \forall n \\ 0 \leq & u_{nk}, t_{nk} \leq 1, \quad \forall k, n.\end{aligned}$$

If $d^2_{\mathbf{A}_k nk} > 0$ for all k, n, m, η, and **X** contains at least K distinct observations then (3.44) is minimized only if condition (3.42) is fulfilled in addition to [36]:

$$t_{nk} = \frac{1}{1 + \left(\frac{b}{\gamma_i} d^2_{\mathbf{A}_k nk}\right)^{1/(\eta-1)}}, \quad \forall k, n \tag{3.47}$$

and

$$\mathbf{v}_k = \frac{\sum_{n=1}^{N} (a u_{nk}^m + b t_{nk}^\eta) \mathbf{x}_n}{\sum_{n=1}^{N} (a u_{nk}^m + b t_{nk}^\eta)}, \quad \forall k. \tag{3.48}$$

The necessary condition in (3.42) for u_{nk} is a function of $\mathbf{x}_n$ and all centroids **V**, whereas the necessary condition for t_{nk} in (3.47) is a function of $\mathbf{x}_n$ and $\mathbf{v}_k$ alone. Hence, u_{nk} is affected by the position of all K clusters, while t_{nk} depends solely on the distance between $\mathbf{x}_n$ and $\mathbf{v}_k$, in addition to the constant γ_k. We therefore regard u_{nk} as the relative typicality, and t_{nk} as the absolute typicality.

The redefinition of the functional in (3.44) causes some interesting properties. For instance, PFCM behaves like FCM when the exponents, m and η, grow without bound. In other words, regardless of the choice of a and b, all K centres will approach the overall mean when $m, \eta \to \infty$. Equation (3.48) indicate that choosing $b > a$ results in the centres being more influenced by the typicality values rather than the membership values. In order to reduce the effect of outliers, one should therefore choose values such that b is greater than a. Similar effects is also applied to the choice of m and η, where choosing $m > \eta$ also reduces the effect of outliers.

3.3.5 Validity Measures of Fuzzy Clustering

In section 3.3.2 we mentioned that different validity indices are used in order to detect the correct number of clusters K. The concept of measuring the degree of similarity within clusters and dissimilarity between clusters is open for interpretation and can be expressed in several ways. Accordingly, different validity measures have been presented in the literature [26, 37]. Partition Coefficient (PC) [38] and Partition Entropy (PE) [39] have historically been the most prominent validity indexes. The former measures how close the fuzzy solution is to the corresponding hard solution, where the hard solution is defined as classifying each observation to the cluster with the highest membership value. The formula for the partition coefficient is given as

$$V_{PC} = \frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} u_{nk}^2,$$

taking values in the range $[1/K, 1]$. A value equal $1/K$ indicates that all membership values are $1/K$. The latter measures the amount of overlap between clusters, and is defined as

$$V_{PE} = -\frac{1}{N} \sum_{k=1}^{K} \sum_{n=1}^{N} u_{nk} \log u_{nk}.$$

This index takes values in the range $[0, \log K]$, where a value close to $\log K$ indicates an absence of any clustering structure in the data, or the fitted clustering algorithm has been unsuccessful in unravelling it. We therefore seek the parameter values which maximizes V_{PC}, and minimizes V_{PE}.

These two indices lacks however a direct connection to the geometrical structure of the data, in addition to often decreasing with the value of K, which causes a disadvantage. Intuitively, clarity and compactness of a classification should increase with the number of classes. The Xie-Beni index [40]

$$V_{XB}(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \frac{\sum_{k=1}^{K} \sum_{n=1}^{N} (u_{nk})^m \|\mathbf{x}_n - \mathbf{v}_k\|^2}{N \cdot \min_{k \neq k'} \left(\|\mathbf{v}_k - \mathbf{v}_{k'}\|^2 \right)}, \tag{3.49}$$

which measures both the compactness and separateness of fuzzy clusters, has been found to perform well in practice. This measure can be interpreted as the ratio between the total within-cluster variation and the separation of the cluster centres. Hence, the optimal value for K can be found by minimizing (3.49). Another popular index was proposed by Fukuyama and Sugeno [41] written as

$$V_{FS}(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \sum_{k=1}^{K} \sum_{n=1}^{N} (u_{nk})^m \|\mathbf{x}_n - \mathbf{v}_k\|^2 - \sum_{k=1}^{K} \sum_{n=1}^{N} (u_{nk})^m \|\mathbf{v}_k - \bar{\mathbf{v}}\|^2, \tag{3.50}$$

where $\bar{\mathbf{v}}$ is the mean of the cluster prototypes. As for V_{XB}, small values of V_{FS} suggest a good partition with compact and well-separated clusters.

These indices are often used to establish the optimal value of K, but also in the selection of other tuning parameters. Despite choosing an appropriate clustering algorithm, improper values of the algorithmic parameters will cause for partitions that does not reflect the desired clustering of the data.

The reason to why the hyperparameter m often is chosen in the interval $[1.5, 2.5]$ is because of its effect on (3.49) and (3.50). As $m \to 1$ from above, the partition becomes hard and $\mathbf{v}_k$ will then be ordinary means of the clusters. Here, the Xie-Beni index will still work well, but the

Fukuyama-Sugeno index will behave like the trace of the within-cluster scatter matrix, which is not a desirable feature. On the other hand, as $m \to \infty$, the partition becomes completely fuzzy which means that all $\mathbf{v}_k$'s will be equal to the mean of $\mathbf{X}$. The limit of the Xie-Beni index will also tend to infinity which causes instability, while the Fukuyama-Sugeno index might either take an indeterminate form or a zero index value. Hence, neither of them will be capable of discriminating the number of clusters.

Chapter 4

Experiments and Analysis

The purpose of this chapter is to present and explain the results found using the models given in chapter 3. First, the process of modifying the data is briefly explained. Second, several logistic-, support vector machine- and fuzzy clustering models have been tested and evaluated. A summary of performance used in a comparison of the supervised models is given in Table B.1 in appendix B.

All model fitting and diagnostics are done using **R**. The total dataset is made up of account information from 40639 distinct collection cases. When training the models 75% of the dataset is used, and the remaining 25% is used in model validation.

4.1 Data Preparation

As mentioned previously, the models are going to predict the RR based on 12 months of observations. The covariates included in the initial dataset provided by Sparebank 1 are given in Table A.1 in appendix A. We note that all transactional variables have one distinct value each month, resulting in 12 variables describing the same type of behaviour. We have for example that ClosingBalance1, ClosingBalance2, etc. correspond to the closing balance $1, 2, ...$ months before the account is sent to collection. One important observation is that in 19% of the collection cases is the age of the account lower than 12 months, and it is reason to believe that the behaviour of these customers is somewhat different than from the older customers. Furthermore, since several covariates will be non-existent for these newer accounts as a 12 month history will not be present, two different models will be built: one for accounts newer than six months, and one for accounts older than six months. This is done by dividing the dataset into `data.new` and `data.old`. The resulting response balance for the two datasets is reported in Table 4.1, where we observe that the dataset consisting of the older accounts have a higher fraction of recovering accounts.

In order to manage the missing transactional covariates present for accounts newer than 12 months and the high number of variables present in both datasets, new covariates with the

Table 4.1: Accounts divided into recovering (RR= 1) and non-recovering (RR= 0) for `data.new` **(a)** and `data.old` **(b)**

(a)

RR	Frequency	Percentage
1	2237	60.02
0	1490	39.97

(b)

RR	Frequency	Percentage
1	29287	79.64
0	7488	20.36

aim of summarizing the behavioural trends were created, and are presented in Table A.2 in the appendix A. The hope is that these new variables are able to capture the variability present in the original variables. These are variables such as the maximum amount of withdrawn cash, how many months the account has had status *normal* instead of *payment reminder* or *collection warning sent*, the maximum closing balance, average cash withdrawals, maximum cash withdrawals, maximum cash transfers, average number of active months were the payment has been lower than 5% of the closing balance, closing balance the first active month, the number of overdrafts, and a flagging variable which is raised if a customer who has ignored one payment suddenly is able to pay later. Note that in order to obtain an accurate representation of average usage, another covariate denoted Active was created. Active is the number of months there has been activity on the credit card, and so the covariates describing average behaviour is only for these active months. All summarizing covariates, except for incurred interest, are divided by the credit limit as the spending amount is correlated with the credit limit.

The final correlation matrix for `data.new` is presented in Figure 4.1. We notice that the correlation between MaxClosingBalance and ClosingBalanceCollectionWarning is $\rho = 0.99$, and the correlation between DaysSinceLastTime and Recurring is $\rho = -0.95$. The predictors MaxClosingBalance and DaysSinceLastTime are therefore removed for models of the newer accounts.

Some of the predictors generated to fit the dataset `data.new` are not included for `data.old`. These are predictors which describe behaviour during the time after the opening of the account. The correlation plot for the included predictors in `data.old` is given in Figure 4.2, where we notice the high correlation between MaxClosingBalance and ClosingBalanceCollectionWarning. As for the previous dataset, we remove MaxClosingBalance.

Finally, the variable SerialN which are two of the digits of the social security number is mainly just information regarding which country the customer is from. So, even though the covariate is given as a number, it can actually be considered as a factor. As the numbers range between $00 - 49$, we are faced with a great number of parameter estimates. This covariate is therefore divided into four different levels based on the first number of the two digits, i.e $0 - 4$, to facilitate this problem.

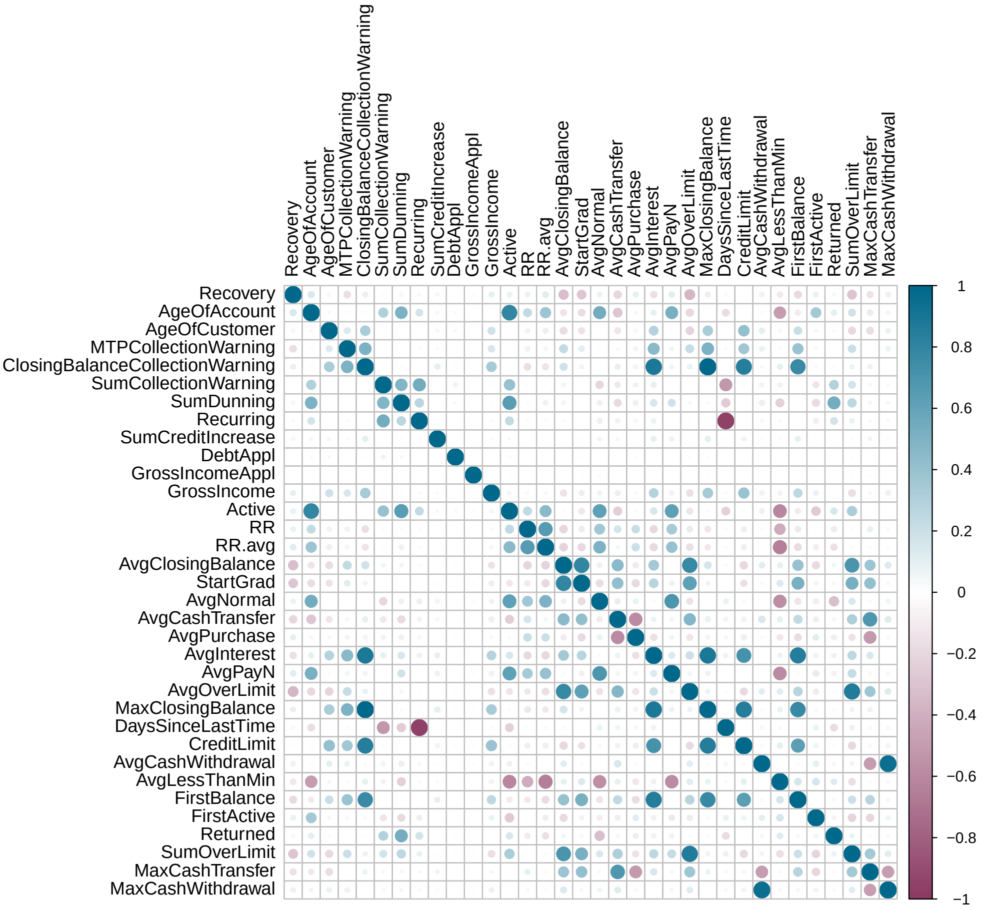


Figure 4.1: Symmetric correlation matrix for quantitative variables in `data.new`. Darker blue dots or darker purple dots correspond to highly correlated variables and larger dots correspond to higher absolute correlation.

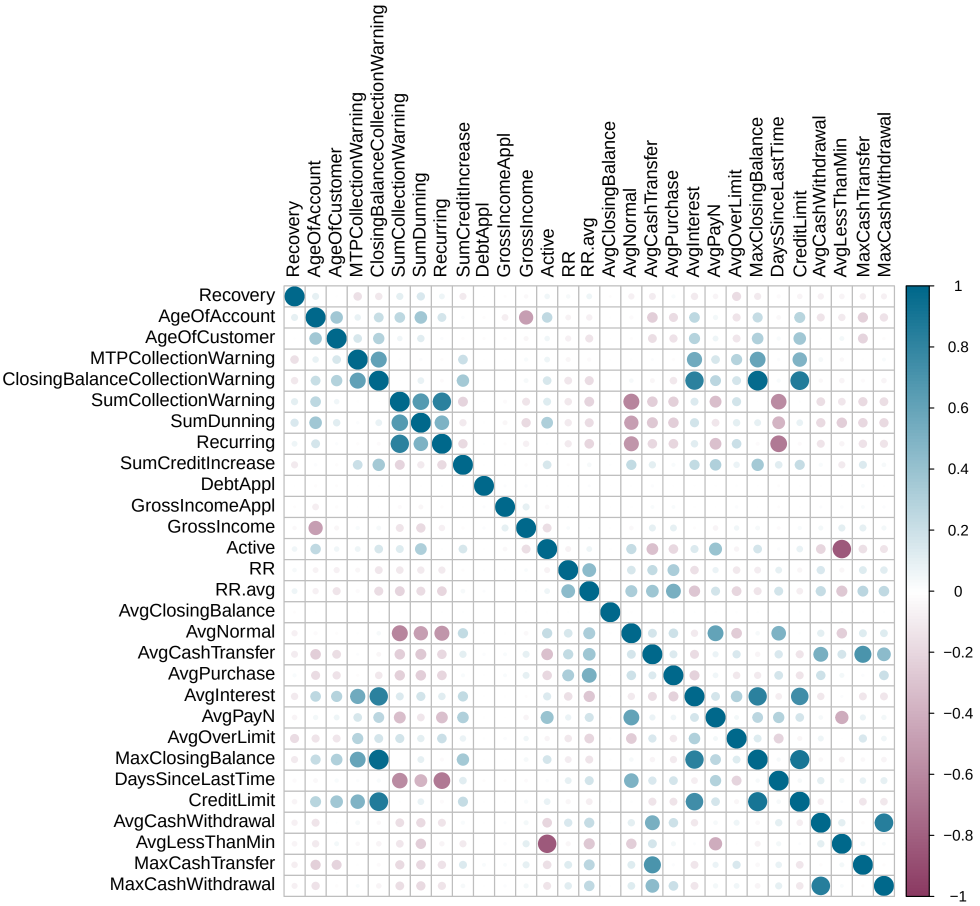


Figure 4.2: Symmetric correlation matrix for quantitative variables in `data.old`. Darker blue dots or darker purple dots correspond to highly correlated variables and larger dots correspond to higher absolute correlation.

4.2 Logistic Model Analysis

4.2.1 Model for New Accounts

Model Selection

We start by fitting the full model consisting of 73 coefficients using the function glm in the package **stats**. Interactions are not included due to the high number of available explanatory variables. The summary of the full model is given in appendix B, where we notice that several covariates are not statistically significant. Hence, a reduction is desired, which can be performed through both a backward elimination as well as regularization. The step function applies the concept of backward elimination based on the AIC. The output of the reduced model using this function is given in appendix B. An ANOVA test for comparing the full- and reduced model using the likelihood ratio as the statistic is given in appendix B. With a p-value $= 0.6926$, we conclude that the full model is not significantly better than the reduced model.

As mentioned in section 3.1.3 a discrete way of performing model selection often produces estimates exhibiting high variance. we will, therefore, also perform ridge regression, the lasso, and elastic net regularization on the dataset. The package **glmnet** is used to perform ridge, lasso and elastic net regularization on the training set of `data.new`. Applying lasso, ridge and elastic net with $\alpha = 0.5$ generates the coefficient profiles presented in Figure 4.3. In the top panel, each curve corresponds to the lasso-, ridge- and elastic net coefficient estimate for each of the 77 variables plotted as a function of $\log \lambda$. At the extreme left side of the plot, λ is essentially zero, which corresponds to a coefficient estimate equal to the usual log-likelihood estimate. As λ increases, the estimates shrink towards zero. The bottom panel shows which values of $\log \lambda$ that correspond to the minimum mean squared error.

Note that since our design matrix includes qualitative variables, we are transforming these into dummy variables. In order to detect which value of α yields the best model selection, 11 different models were fitted using values of α in the range 0 to 1. Each model is fitted for an automatically selected range of λ values, which can be seen in Figure 4.3, and the optimal model for each α is chosen based on the lowest value of MSE. The MSE for different values of α is plotted in Figure 4.4.

For the newer accounts, an optimal value of α was calculated to be $\alpha_{\text{new}} = 0.7$, which indicates a heavier weighting of the lasso. Utilizing this model suggests a reduction of 78% of the covariates, which is very high. This indicates that the variation in all of these covariates was relatively small, or correlated with the remaining variables, and had therefore little or no influence on the response. The included covariates with their respective estimate are presented in Table 4.2, where we observe that 16 variables are included in the final model. Furthermore, we will inspect the estimates obtained through the backward elimination procedure.

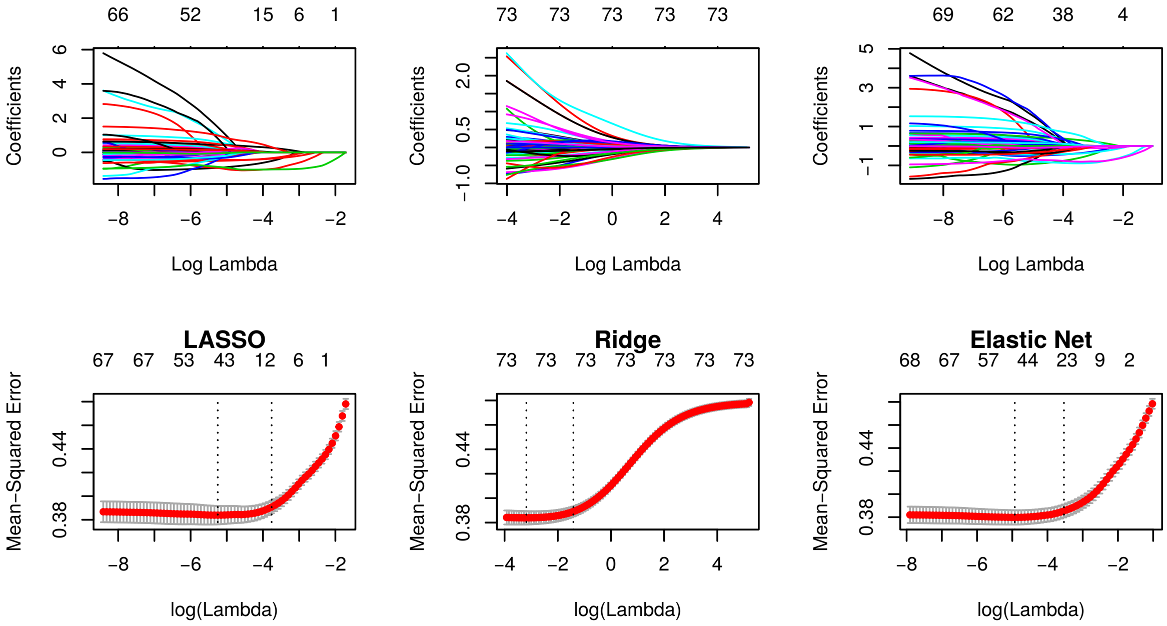


Figure 4.3: Lasso, Ridge and Elastic Net on `data.new`. The standardized coefficients (top) and mean squared error (bottom) are shown as a function of $\log(\lambda)$.

Parameter Analysis

We notice that 34 out of the 73 initially included covariates remains in the reduced model obtained from a backward elimination procedure, whereas 18 of them are not considered statistically significant at a 10% significance level. We notice that the backward elimination procedure includes more covariates compared to the regularization approach. When the variables in a regression model are highly correlated, their coefficients will become poorly determined and exhibit high variance. For instance, a large coefficient for one variable can be cancelled by an equally negative coefficient on a correlated variable. Imposing the penalization in ridge, lasso and elastic net, alleviate this problem. In Figure 4.1 we observed that several covariates were highly correlated, which explains the high reduction from the elastic net approach. Among the included variables in the regularized model, five of them are not included in the reduced model. These are ProductSB1 EXTRA MC, AvgClosingBalance, AvgLessThanMin, FirstActive and MaxCashWithdrawal. In order to obtain some notion of how the regularized model explains the variability in the data compared to the reduced model, we fit a logistic model by only including the variables remaining for the regularized model, and investigate their significance. The model output is given in appendix B, and reveal that only the predictor AvgLessThanMin is not significant at a 10% significance level. By performing a likelihood ratio test on the reduced model including 34 predictors, and the regularized including 16 predictors, we obtain a p-value $= 4.958 \cdot 10^{-6}$. This tells us that the reduced model gives a better fit than the regularized, and so our model analysis will primarily concern the reduced model.

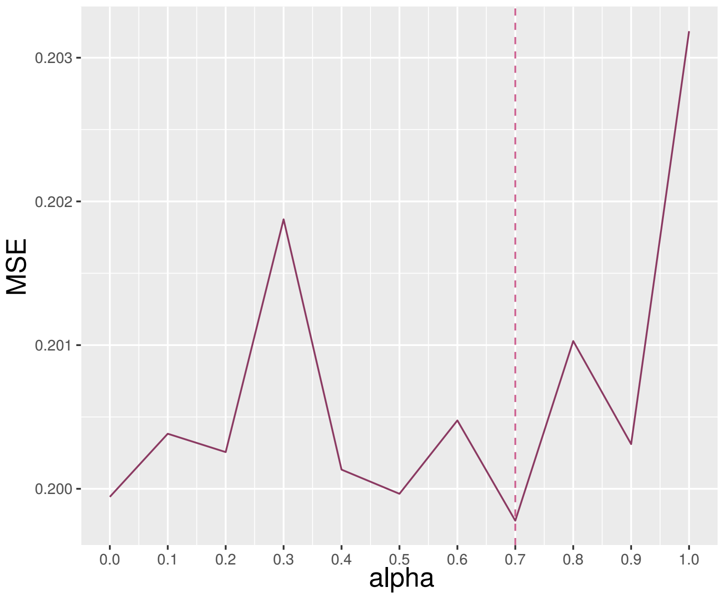


Figure 4.4: MSE as a function of α fitted for `data.new`

Comparing the effects in the reduced model, we see that AgeOfAccount is considered highly significant even though it is not included in the regularized model. The estimate of AgeOfAccount is calculated to be $\hat{\beta}_{\text{AgeOfAccount}} = 0.2$, with the multiplicative effect of $\exp(\hat{\beta}_{\text{AgeOfAccount}}) = 1.221$. Thus, increasing the age of the account with one month leads to a multiplication of the odds of the RR by 1.221. The average number of payments, AvgPayN, and the average number of times there have been an overdraft on the account, AvgOverLimit, are also considered important. The estimate of AvgPayN is $\hat{\beta}_{\text{AvgPayN}} = 1.547$, which yields a multiplication of the odds by $\exp(\hat{\beta}_{\text{AvgPayN}}) = 4.697$. This suggests that customers transferring money to their credit card frequently have the tendency to recover from collection, opposed to customers only transferring money once or maybe never. The estimate of AvgOverLimit is $\hat{\beta}_{\text{AvgOverLimit}} = -1.096$, equivalent to a multiplicative effect of $\exp(\hat{\beta}_{\text{AvgOverLimit}}) = 0.334$. Hence, increasing the rate of overdrafts produces a lower probability of recovering. One interesting observation is the negative estimate of AvgCashTransfer, which suggests that customers which have the habit of transferring money from the credit card to other bank accounts are at greater risk of not recovering.

When studying the effect of each predictor it is important to remember what type of model we are considering. The estimated effects are the effects of one predictor given that the other predictors are fixed. For a logistic model where the relationship between neither the probability nor the odds is linear, the effect of a predictor is dependent on the fixed level of the other predictors. This complication is visualized in Figure 4.5 where an equal increase in the linear predictor gives a different change in the probability. In other words, the effect of one predictor is smaller if $p = \pi$ is close to either 0 or 1, than when p is close to 0.5. Instead of only looking at the estimated effect one could consider the maximum effect which is achieved for a p around

Table 4.2: Predictors included in the model with $\alpha = 0.7$ for the dataset `data.new`

Predictor	$\hat{\beta}$	$\exp(\hat{\beta})$
DebtCollectionCompanyGOT	$9.14 \cdot 10^{-2}$	1.096
ProductSB1 EXTRA MC	$-2.11 \cdot 10^{-1}$	0.810
SerialN10	$-3.52 \cdot 10^{-1}$	0.703
SerialN20	$-2.52 \cdot 10^{-1}$	0.777
RecruitmentChannelOperatoerkanal	$2.87 \cdot 10^{-1}$	1.332
MTPCollectionWarning	$-5.90 \cdot 10^{-5}$	0.999
GrossIncome	$4.74 \cdot 10^{-7}$	0.999
EvaluationMethodOrdinaer	$-4.29 \cdot 10^{-1}$	0.651
AvgClosingBalance	$-7.91 \cdot 10^{-1}$	0.453
AvgInterest	$-3.93 \cdot 10^{-4}$	0.999
AvgPayN	$6.26 \cdot 10^{-1}$	1.870
AvgOverLimit	$-9.79 \cdot 10^{-1}$	0.376
AvgLessThanMin	$-2.23 \cdot 10^{-1}$	0.800
FirstActive	$7.52 \cdot 10^{-2}$	1.078
Returned	$9.67 \cdot 10^{-1}$	2.630
MaxCashWithdrawal	$-1.51 \cdot 10^{-1}$	0.860

0.5. The maximum effect of, say RR.avg which is the average recovery rate each active month, is calculated to be $\Delta p^{+} = 0.599$. This implies that increasing the average recovery rate for each month by one, could increase the probability of recovery with 0.599. Similarly, the minimum effect is calculated to be $\Delta p^{-} = 0.047$. The difference between Δp^{+} and Δp^{-} illustrates how one predictor could contribute highly to the probability, and sometimes could be considered irrelevant.

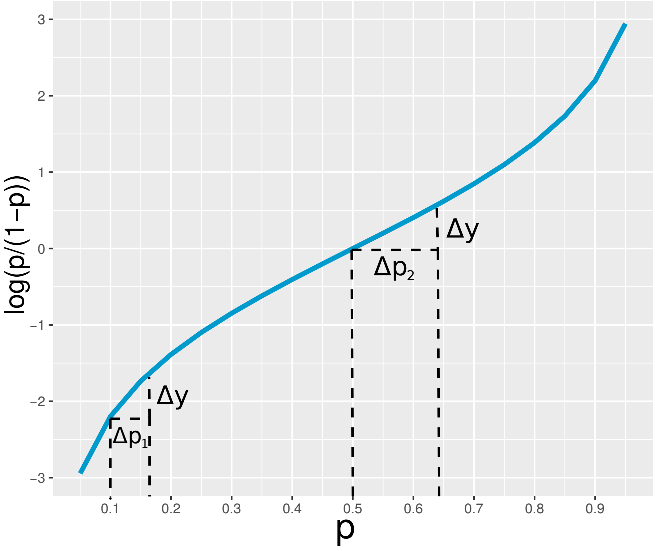


Figure 4.5: Illustration of how the effect of one predictor is dependent on the fixed values of the other predictors. The change Δp is small for p close 0 and 1, and high for p close to 0.5

Predictive Performance

It is now interesting to investigate how well this model predicts newly defaulted accounts, in addition to comparing the predictive performance of the reduced- and regularized model. Figure 4.6 shows the density plots for the predicted probabilities for both recovering and non-recovering accounts given from the reduced model.

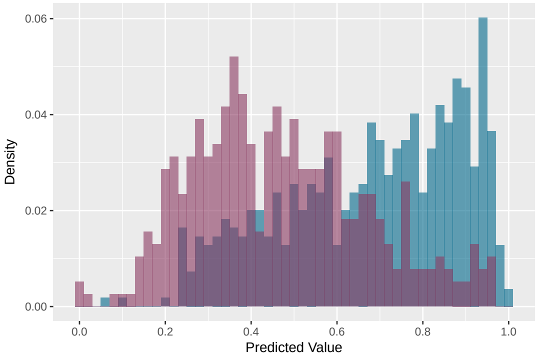


Figure 4.6: Distribution of the predicted values for both recovered (blue) and non-recovered (purple) accounts in the reduced logistic regression model for `data.new`

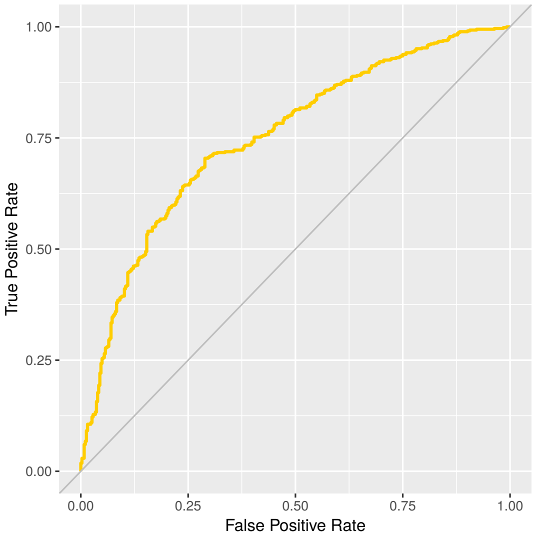


Figure 4.7: ROC curve of reduced logit model for `data.new`

Intuitively, we let the threshold of the decision be $\hat{\pi} = 0.5$, which means that a predicted probability above 0.5 gives $\text{RR} = 1$. Using the package **caret** we obtain the confusion matrix for the reduced model which is given in Table 4.3. Since we have divided the data such that the test set is used in model validation, cross-validation is not necessary. We calculate the accuracy to be $AC = 0.679$ and the precision to be $P = 0.710$, which can be seen in Table B.1 in appendix B. The ROC curve is given in Figure 4.7, and the area under the curve is computed using the **pROC** package to be $\text{AUC} = 0.751$. As this is actually not a binary problem, we set a lower and an upper threshold of how certain we want to be in our decision. The boundary 0.5 is quite crisp, and will not catch the accounts which can be placed in the middle, i.e accounts with $0 < \text{RR} < 1$. From Figure 4.6 we see that the two distributions overlap for nearly all predictive values. However, in order to increase the certainty of our decision, we let the lower and upper threshold be 0.2 and 0.8, respectively. Applying these new limits yields $AC = 0.866$ and $P = 0.864$, which is notably better. It is also worth noticing that 59.7% of the accounts in the test set were placed in the uncertain category.

Some of the most prominent covariates for `data.new`, according to the elastic net approach, is if the evaluation method were of type *ordinary*, the average number of payments each month, if the recruitment channel was *operational channel* and the average number of times the account overdrafts each active month.

Table 4.3: Confusion matrices of the reduced logit model for `data.new` with threshold $\hat{\pi} = 0.5$ (left), and $\hat{\pi} \in [0.2, 0.8]$ (right).

		Prediction	
		False	True
Reference	False	212	172
	True	127	421

		Prediction		
		False	F/T	True
Reference	False	36	316	32
	True	5	340	203

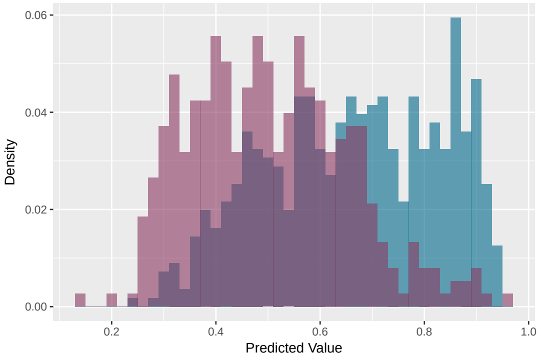


Figure 4.8: Distribution of the predicted values for both recovered (blue) and non-recovered (purple) accounts in the regularized logistic regression model for `data.new`

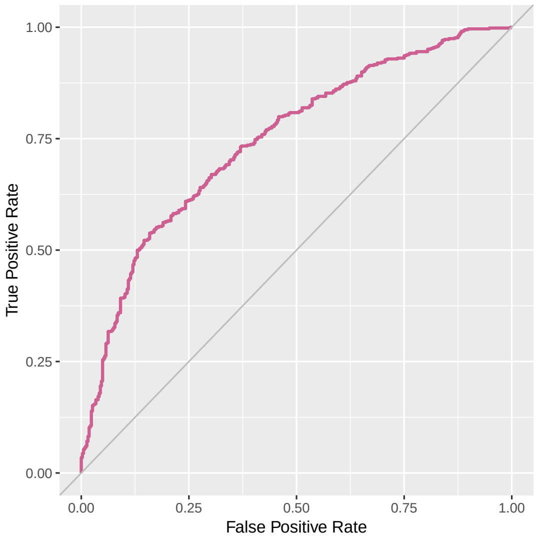


Figure 4.9: ROC curve of logit model with elastic net regularization for `data.new`

We see from Figure 4.8 that the distribution of the predictions is somewhat different from the reduced model. The overlap is evident but is now more centred towards the interval $[0.4, 0.7]$. Hence, in addition to addressing the predictive performance for the crisp boundary at 0.5, we add a lower and an upper threshold of 0.3 and 0.75, respectively. We obtain the confusion matrix given in Table 4.4 for the regularized model.

Table 4.4: Confusion matrices of the model obtained using elastic net regularization for `data.new` with threshold $\hat{\pi} = 0.5$ (left), and $\hat{\pi} \in [0.3, 0.75]$ (right).

		Prediction	
		False	True
Reference	False	201	176
	True	114	441

		Prediction		
		False	F/T	True
Reference	False	26	329	22
	True	3	359	193

Using the crisp decision boundary yields $AC = 0.689$ and $P = 0.796$, which is actually higher than for the reduced logistic model. The ROC curve is presented in Figure 4.9, and the area under the curve is computed to be AUC $= 0.746$. Imposing the same lower and upper limit as for the reduced model results in an accuracy of $AC = 0.898$ and a precision of $P = 0.898$, which is an improvement from the crisp boundary decision. However, 73.8% of the accounts in the test set were placed in the uncertain category, which also could be the reason for the high accuracy and precision.

Using cross-validation we want to explore if the two models are significantly different at predicting. We use the boundary $\hat{\pi} = 0.5$ when calculating the precision of the reduced and regularized model. We let the difference on fold i be $p_i = p_i^1 - p_i^2$, and execute a 10-fold CV paired t-test. We calculate the statistic to be $T = -1.304$, and by letting $\alpha = 0.05$ we have the critical value $t_{0.025,9} = 2.26$. We can, therefore, conclude that the reduced model has not a significantly better precision than the regularized model. We can now do the same with the accuracy, which gives an observed value $T = 0.587$, also not rejecting the null hypothesis stating that their accuracy is equal. Thus, by using these two measures one cannot conclude that either model is better than the other.

Goodness of Fit

Now, we want to assess the goodness of fit of the reduced model first through the McFadden R-squared, which is found using the package **BaylorEdPsych**. The value is calculated to be $R^2_{\text{McFadden}} = 0.19$, which as mentioned in section 3.1.4 indicate an acceptable model fit.

Second, performing the Hosmer-Lemeshow test using the package **ResourceSelection** with

$$g = \max\left(10, \min\left\{\frac{1689}{2}, \frac{2795-1689}{2}, 2+8\left(\frac{2795}{1000}\right)^2\right\}\right) \approx 65$$

results in the output given in appendix B. With a p-value = 0.4023 we have no evidence to reject the null hypothesis stating that the model is correctly specified. The plot of Cook's distance is presented in Figure 4.10, where we see that there are a few observations which attain a higher influence than the rest. According to section 3.1.4 the critical leverage value is $h_{ii} = 3 \times 35/2795 = 0.038$. There are 50 observations in the training set which have a leverage value higher than this boundary. Removing these observations results in a lower AIC $= 3051.1$ for the reduced model. However, it is more interesting to investigate if the prediction is improved. The accuracy and precision are calculated to be $AC = 0.685$ and $P = 0.716$, barely improved from the reduced model.

The residual deviance for the reduced model is calculated to be $D = 3105.6$ on 2761 degrees of freedom, which could indicate lack of covariates, power, interactions terms, or that the data should be grouped. we will, therefore, continue in section 4.3.1 by looking at the support vector machine approach.

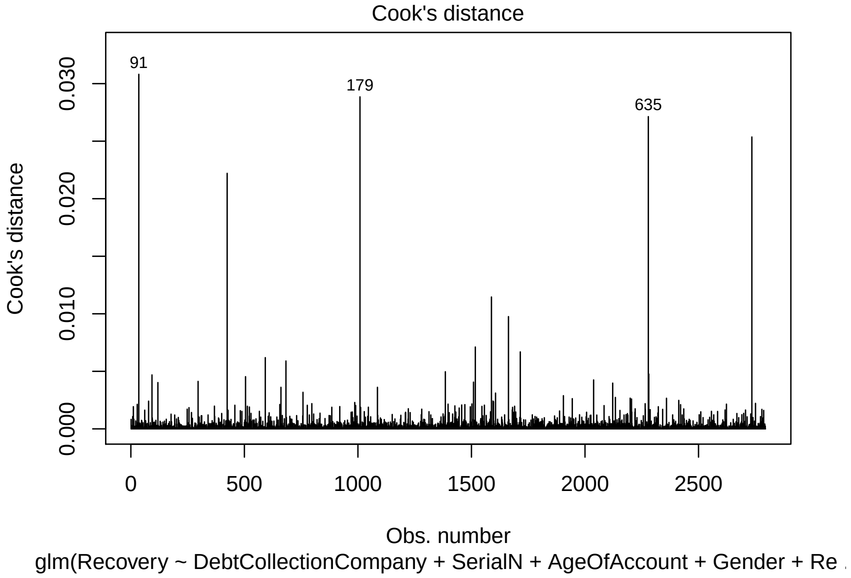


Figure 4.10: Plot of Cook's distance for `data.new`

4.2.2 Model for Old Accounts

Model Selection

The summary of the model fitted by including all 77 covariates is given in appendix B. We observe that there are some covariates which are not statistically significant. Hence, a reduction through a backward elimination based on the AIC is performed. The summary of the reduced model, as well as the output from the ANOVA test between the full- and reduced model are also given in appendix B. We have a p-value $= 0.6956$, which results in the full model not being significantly better than the reduced model. We will, therefore, continue with analyzing the reduced model instead of the full model.

A reduction through the use of ridge regression, the lasso and elastic net is also done for the older accounts and is presented in Figure 4.11. Similarly, as for the newer accounts, the optimal value of α is selected by fitting 11 models using values of α in the range 0 to 1. The MSE for the different models is plotted in Figure 4.12.

The optimal value of α was found to be $\alpha_{\text{old}} = 0.9$, which implies a heavy weighting of lasso regression. Fitting this model reduced the number of covariates with 80.8%, which is consistent with the high correlations observed in Figure 4.2. Also, considering the results obtained for the newer accounts this is not surprising. If a covariate has low variance among the newer accounts, it is expected a somewhat similar result for the older accounts. However, the difference in α_{new} and α_{old} suggests that several covariates have a lower influence on older accounts than newer ones. The included covariates, as well as their estimates, are presented in Table 4.5, but without a measure of their significance, we cannot accurately evaluate their importance. A logistic model only including the variables preserved by the elastic net approach

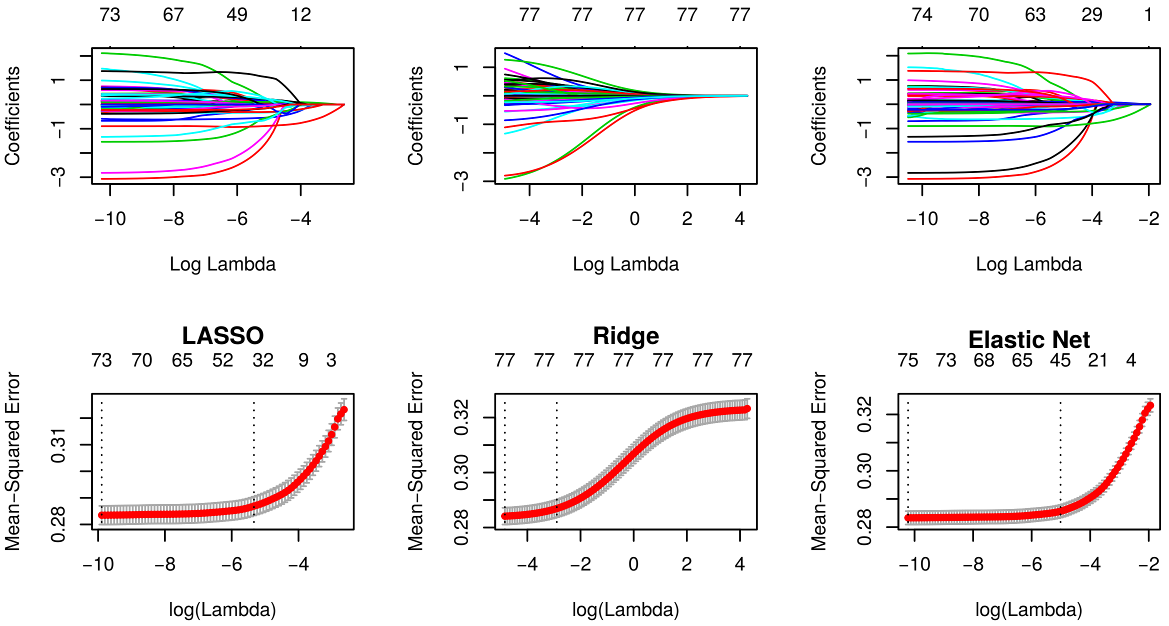


Figure 4.11: Lasso, Ridge and Elastic Net on `data.old`. The standardized coefficients (top) and mean squared error (bottom) are shown as a function of $\log(\lambda)$.

is fitted. This model reveals that all 14 predictors are considered significant at a 1% significant level. Only the average amount of transactions to other bank accounts, AvgCashTransfer, is included in the regularized model and not the reduced model. Performing a likelihood ratio test on the two models, we get a p-value $< 2.2 \cdot 10^{-16}$ stating that the reduced model is significantly better than the regularized.

Parameter Analysis

The backward elimination procedure yields a model which includes 70 out of the 77 initially included covariates, whereas 21 of them are not considered statistically significant at a 10% significance level. The elastic net, however, only preserves 14 covariates. Several predictors have established themselves as important in LGD/RR modelling in literature. Bellotti and Crook (2012) [42] demonstrated that time with bank, which is here denoted AgeOfAccount, has significantly positive effect on the RR. They also found that balance at collection had a negative impact on the RR. The results obtained for both the new- and old accounts model are consistent with this result. We have also included predictors which have not been tested in previous literature. The average recovery rate for the previously active months, RR.avg, is an important feature in the older account, with a multiplicative effect of $\exp(\hat{\beta}_{\text{RR.avg}}) = 3.146$. The multiplicative effect is lower for the older accounts compared to the newer accounts, indicating that it is more important for newly joined customers to pay down their monthly invoices in order to recover from default. The predictor AvgNormal is also considered prominent with a multiplica-

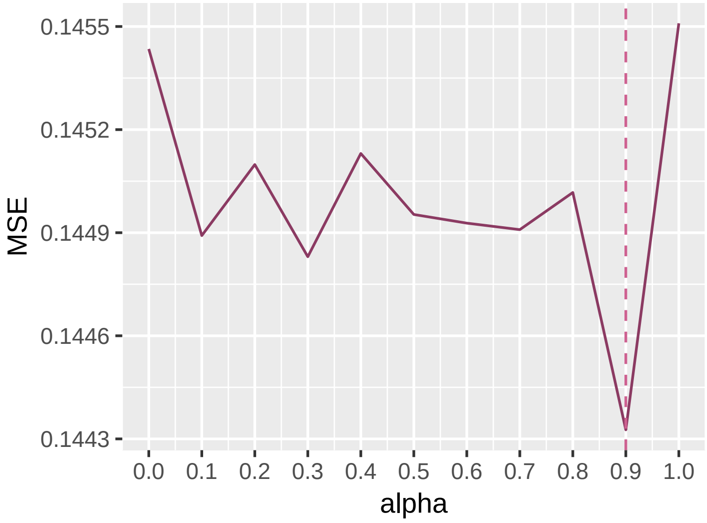


Figure 4.12: MSE as a function of α fitted for `data.old`

tive effect of $\exp(\hat{\beta}_{\text{AvgNormal}}) = 0.251$. The negative effect of AvgNormal, which suggest that the average number of times the account has had status *normal* causes a decrease in the recovery rate, is quite surprising. Intuitively one might think that an account that has had a normal behaviour except for a few months would have a greater chance of recovery. This is however not the case. One possible explanation could be just plain laziness. Meaning there exist a great deal of customers who are able to pay, but not willing. Another reason could be that some customers ignore invoices with status ”normal” due to poor finance, and only pays the invoices with status *payment reminder* in fear of ending up in default. Once they end up in default they will find the means to pay the minimum amount required. In that way will lower risk customers have the behaviour of fluctuating between *normal* and *payment reminder*, which yields a relatively high rate of status *normal*. From the data it is found that the average value of AgeOfAccount for recovering accounts is higher compared to non-recovering accounts. Combining this with the positive effect of AgeOfAccount, it is not as astonishing that an older account have an increased number of payment reminders compared to newer accounts. Additionally, several predictors which were not considered significant in the newer account model are included in the model for the older accounts. These are covariates such as the average amount of active months where the sum of payments were lower than 5% of the closing balance, the maximum amount of transfers to other accounts, and the maximum amount of cash withdrawn.

Table 4.5: Predictors included in the model with $\alpha = 0.9$ for the dataset `data.old`

Predictor	$\hat{\beta}$	$\exp(\hat{\beta})$
ProductSpareBank 1 MasterCard Gold	$4.98 \cdot 10^{-2}$	1.051
SerialN10	$-2.84 \cdot 10^{-1}$	0.753
SerialN20	$-2.22 \cdot 10^{-1}$	0.801
SerialN40	$1.76 \cdot 10^{-3}$	1.002
MTPCollectionWarning	$-5.79 \cdot 10^{-5}$	0.999
SumDunning	$9.59 \cdot 10^{-2}$	1.101
SumCreditIncrease	$-5.03 \cdot 10^{-2}$	0.951
PaymentTypePrint	$7.46 \cdot 10^{-2}$	1.077
EvaluationMethodOrdinær	$-2.12 \cdot 10^{-1}$	0.809
RR.avg	$1.33 \cdot 10^{-1}$	1.142
AvgNormal	$-6.89 \cdot 10^{-2}$	0.933
AvgCashTransfer	$-5.34 \cdot 10^{-2}$	0.948
AvgInterest	$-5.54 \cdot 10^{-5}$	0.999
AvgOverLimit	$-7.66 \cdot 10^{-1}$	0.465

Predictive Performance

Furthermore, a predictive analysis is performed. We want to examine how well both the reduced- and regularized model predicts, before comparing the two models. We let the initial threshold of our decision be $\hat{\pi} = 0.5$, which results in the prediction from the reduced model given to the left in Table 4.6. The ROC curve is presented in Figure 4.14, where the area under the curve is calculated to be AUC $= 0.737$.

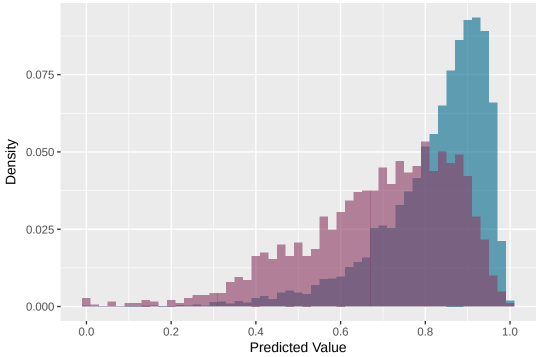


Figure 4.13: Distribution of the predicted values for both recovered (blue) and non-recovered (purple) accounts in the reduced logistic regression model for `data.old`

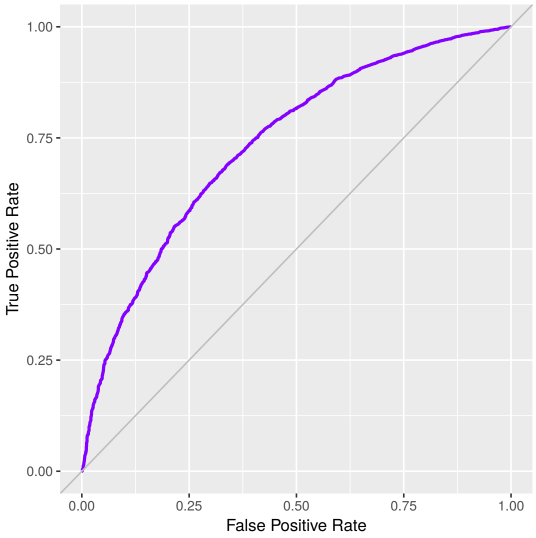


Figure 4.14: ROC curve of reduced logit model for `data.old`

The accuracy and precision are calculated to be $AC = 0.802$ and $P = 0.815$, which is relatively high. However, we can observe that the values False Negative and False Positive compared to True Negatives are relatively high. Consequently, we divide the prediction into three instead of two: certain negative, certain positive and uncertain. From Figure 4.13 we notice that the overlap of the predictive distributions of the two classes are skewed towards probability values around 0.8, which reveals why we obtain such a high value of FP when using a boundary of 0.5. We let the lower and upper threshold be 0.3 and 0.85, respectively. Imposing these thresholds results in an updated accuracy of $AC = 0.905$ and precision $P = 0.909$. Albeit improving the prediction, the size of False Negative and False Positive are still too high compared to the size of True Negative. Here, 53.22% of the observations were placed in the uncertain category.

Table 4.6: Confusion matrices of the reduced logit model for `data.old` with threshold $\hat{\pi} = 0.5$ (left), and $\hat{\pi} \in [0.3, 0.85]$ (right).

		Prediction	
		False	True
Reference	False	293	1604
	True	214	7083

		Prediction		
		False	F/T	True
Reference	False	48	1462	387
	True	23	3431	3843

For the regularized model we obtain the prediction distribution given in Figure 4.15, and ROC curve in Figure 4.16 with AUC $= 0.733$. From the confusion matrix presented in Table 4.7 the accuracy is computed to be $AC = 0.799$, and precision $P = 0.809$, which is similar to the reduced model. The value of False Negative and False Positive is still too high compared to the True Negative. We impose the two thresholds of 0.3 and 0.85. Placing 56.06% observations in the uncertain category gives an updated accuracy of $AC = 0.908$ and precision $P = 0.911$, which is an improvement. However, we still have a high share of misclassified observations. Hence, we need a model which captures the behaviour of high-risk cases better.

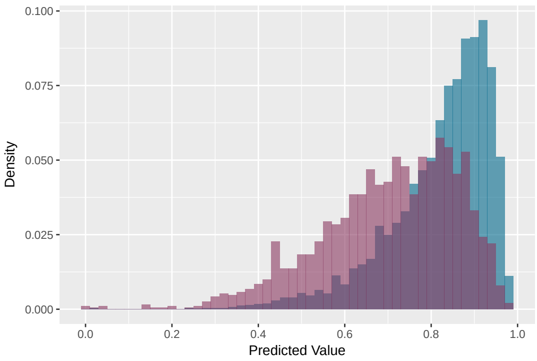


Figure 4.15: Distribution of the predicted values for both recovered (blue) and non-recovered (purple) accounts in the regularized logistic regression model for `data.old`

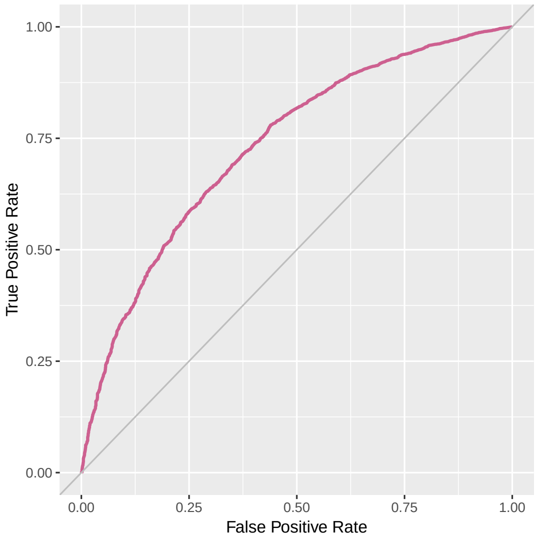


Figure 4.16: ROC curve of logit model with elastic net regularization for `data.old`

Table 4.7: Confusion matrices of the model obtained using elastic net regularization for `data.old` with threshold $\hat{\pi} = 0.5$ (left), and $\hat{\pi} \in [0.3, 0.85]$ (right).

		Prediction	
		False	True
Reference	False	218	1679
	True	172	7125

		Prediction		
		False	F/T	True
Reference	False	24	1517	356
	True	14	3637	3646

We want to assess the difference in prediction through the approach of cross-validation. Letting the boundary be $\hat{\pi} = 0.5$, and executing a 10-fold CV paired t-test using $p_i = p_i^1 - p_i^2$ as the difference in precision for the reduced and regularized model on fold i, results in $T = 11.756$. With a confidence level of 95% we have the critical value $t_{0.025,9} = 2.26$, which results in the rejection of the null hypothesis stating that their precision is equal. Hence, the reduced model is significantly better than the regularized. Similarly, for the accuracy we obtain $T = 1.131$, which do not reject the null hypothesis. This result indicates that the reduced model is better at detecting the positive cases, but overall are they not different.

Goodness of Fit

Finally, we look at the goodness of fit of the reduced model. The McFadden R-squared is calculated to be $R^2_{\text{McFadden}} = 0.12$, which is not very good. The Hosmer-Lemeshow test with

$$g = \max\left(10, \min\left\{\frac{21990}{2}, \frac{27581 - 21990}{2}, 2 + 8\left(\frac{27581}{1000}\right)^2\right\}\right) \approx 2796,$$

gives the output in appendix B. With a p-value $< 2.2 \cdot 10^{-16}$ we reject the null hypothesis stating that our model is correctly specified. However, our confidence in this measure decreases with N, and for the training set we have $N = 27581$, which is actually above our boundary of 25000 stated in section 3.1.4.

The Cook's distance is plotted in Figure 4.17, where we notice that some points attain a higher influence than others. The critical leverage value is calculated to be $h_{ii} = 3 \times 71/27581 = 0.0077$, with 1615 observations exceeding this threshold. Removing these observations yields a reduced model with an AIC $= 23085$ indicating a better model performance, but could just be the result of removing such a large amount of observations. Regardless, our main aim is prediction and consequently, we need to assess whether the prediction is improved. With accuracy $AC = 0.802$ and precision $P = 0.815$, there is no improvement and we continue therefore with including all observations.

The residual deviance for the model is calculated to be $D = 24403$ on 27511 degrees of freedom. There is, therefore, no indication of overdispersion. Nevertheless, the model does not seem to explain the variation in the data very well. we will, therefore, in the next sections attempt to fit a model which better demonstrates this variability.

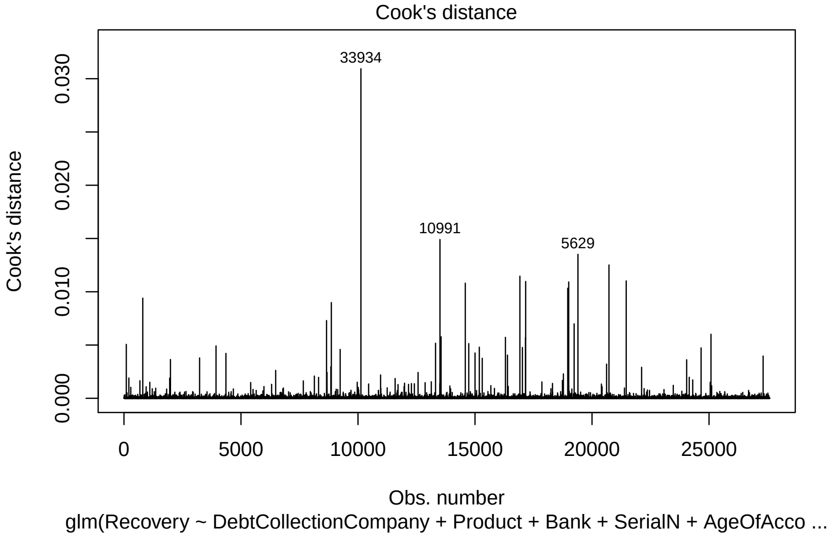


Figure 4.17: Plot of Cook's distance for `data.old`

4.3 Support Vector Machine Analysis

We are using the **kernlab** package and ksvm function when fitting the support vector machine. Aforementioned in section 3.2 typical kernel choices are the Gaussian RBF and Laplace RBF when there is no prior information regarding the data. Both of these includes the two hyperparameters C and σ, requiring some kind of model selection to be done. One approach here is to perform a grid search using cross-validation. This is a straight-forward way of optimizing but comes across as a bit naive. Actually, there exist several more advanced methods which can save computational cost.

One of these methods is known as response surface methodology. The initial values for σ are found using the function sigest. The function estimates a range of values for σ which returns good results based on the 0.1 and 0.9 quantile of $||\mathbf{x} - \mathbf{x}'||^2$. Common choices of C are in the range of $0.1 - 100$ [43]. We will first execute an optimization process of the hyperparameters C and σ with RSS as our objective function using the Gaussian RBF kernel, computed from a 10 fold cross-validation. Furthermore, we are going to fit and analyze the resulting model for each dataset `data.new` and `data.old`.

4.3.1 Model for New Accounts

Model Selection

The initial values are set to $C = \{0.25, 4\}$ and found to be $\sigma = \{4.42 \cdot 10^{-3}, 1.51 \cdot 10^{-2}\}$. The result obtained from performing a 2^2 experiment is presented to the left of Figure 4.18, where we observe an optimal value of RSS $= 5.203 \cdot 10^{-4}$ in the point $(C = 0.25, \sigma = 4.42 \cdot 10^{-3})$. Before following the gradient from the centre of the design space to the first step optimal point, we will investigate whether there is an interaction between C and σ. Assuming the $\log(\text{RSS})$ can be expressed as

$$\log(\text{RSS}) = \beta_0 + \beta_C C + \beta_\sigma \sigma + \beta_{C,\sigma} C \cdot \sigma + \varepsilon, \tag{4.1}$$

we will estimate the value of $\boldsymbol{\beta}$ based on our observations from the 2^2 experiment, where we have

$$\log(\text{RSS}) = \log \begin{bmatrix} 5.203 \cdot 10^{-4} \\ 5.888 \cdot 10^{-4} \\ 6.229 \cdot 10^{-4} \\ 1.528 \cdot 10^{-3} \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_C \\ \beta_\sigma \\ \beta_{C,\sigma} \end{bmatrix} + \varepsilon.$$

The logarithm is chosen in order to have the responses conform more closely to the normal distribution. The estimation of the effects, given in appendix B, were quite similar which could indicate that there exist some interaction between the two hyperparameters. However, Figure B.1 in appendix B reveals that neither one of these estimates are considered significant. Hence, instead of looking at a 2^2 experiment, we will execute a central composite design with $\alpha = \sqrt{2}$, and adjusting the position of one axial point due to the restriction of $C > 0$. Looking at the right plot of Figure 4.18 we obtain a new optimal point in $(C = 4.78, \sigma = 9.75 \cdot 10^{-3})$ with RSS $= 3.203 \cdot 10^{-4}$. Noticing how the optimal point shifts when we instead of a rectangle region, study a circular region, suggest that the RSS is more complex than assumed in (4.1). Hence, we will search around this new optimum in the attempt to find a local optimum. Omitting the calculations, we obtain no further improvement when moving along the gradient from the centre of the design space to the first optimal point. Hence, we perform another CCD around the first optimal point which results in an updated point in $(C = 5.28, \sigma = 4.76 \cdot 10^{-3})$ with RSS $= 2.27 \cdot 10^{-4}$. Moving along the gradient from the newly obtained centre of the design space to the second optimal point does not improve the value of RSS. Hence, we conclude the optimization process stating that we have found a local optimum.

Predictive Performance

The final model is an SVM with the Gaussian RBF kernel and parameter values $C = 5.28$ and $\sigma = 4.76 \cdot 10^{-3}$. Here we have used the option of calculating class probabilities rather than hard classification. The number of support vectors is calculated to be 1888 out of 2795, which accentuates the overlap of observations. This overlap is also visualized in Figure 4.19, which reveals a noticeable amount of overlap for predicted values between 0.3 and 0.7. In other words, accounts which have a very similar behaviour may not have the same recovery rate.

The confusion matrices are given in Table 4.8, which reveal an accuracy of $AC = 0.687$ and precision of $P = 0.704$ when utilizing the crisp boundary $\hat{\pi} = 0.5$. From Figure 4.19 we impose the uncertain category using the boundaries 0.2 and 0.8. We obtain $AC = 0.863$ and $P = 0.858$, which is not superior to the logistic model.

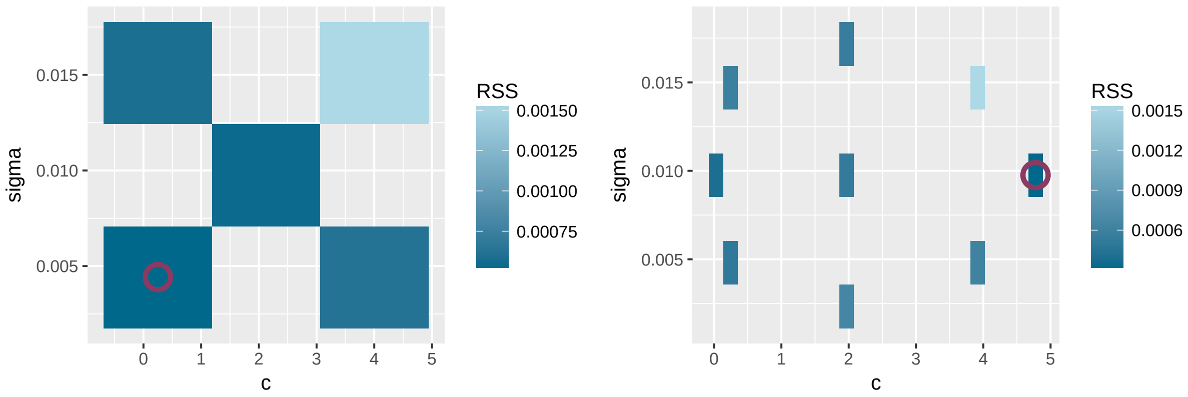


Figure 4.18: Optimization of the hyperparameters C and σ using the concept of RSM for `data.new`. The predictive error is presented in a heat map, and the purple circle denotes the optimal point of (C, σ) for the 2^2 experiment (left) and CCD (right).

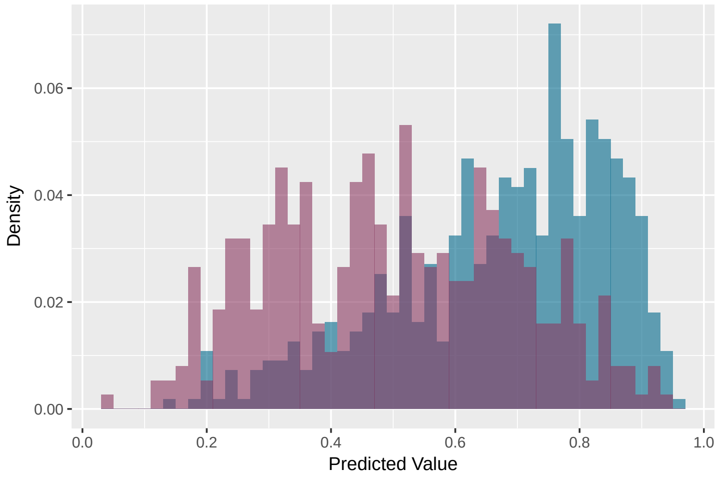


Figure 4.19: Distribution of the predicted values for both recovered (blue) and non-recovered (purple) accounts in the SVM model for `data.new`

Table 4.8: Confusion matrices of the SVM for `data.new` with threshold $\hat{\pi} = 0.5$ (left), and $\hat{\pi} \in [0.2, 0.8]$ (right).

		Prediction	
		False	True
Reference	False	198	186
	True	106	442

		Prediction		
		False	F/T	True
Reference	False	33	321	30
	True	4	363	181

Characteristics of the Support Vectors

Preceding, we want to compare the support vectors with the remaining observations. The predictor StartGrad is computed as the difference in closing balance from the second active month to the first. In other words, how much has the usage increased during the first active month. From Figure 4.20 it is observed that the support vectors for both the low-risk and high-risk cases have a similar distribution with a heavy weight around StartGrad $= 1$. The correctly classified recovering observations have a heavy tail close to StartGrad $= 0$ and implies that these accounts have a lower spending rate the first active month compared to the incorrectly classified observations. Similarly, roughly all the correctly classified non-recovering cases have StartGrad ≥ 1. Hence, the typical difference is that high-risk accounts spend nearly all available credit during the first active month opposed to low-risk accounts which have a much more spread out spending rate. There exist however several recovering accounts which exhibits a typical high-risk spending behaviour, contributing to noise in our model.

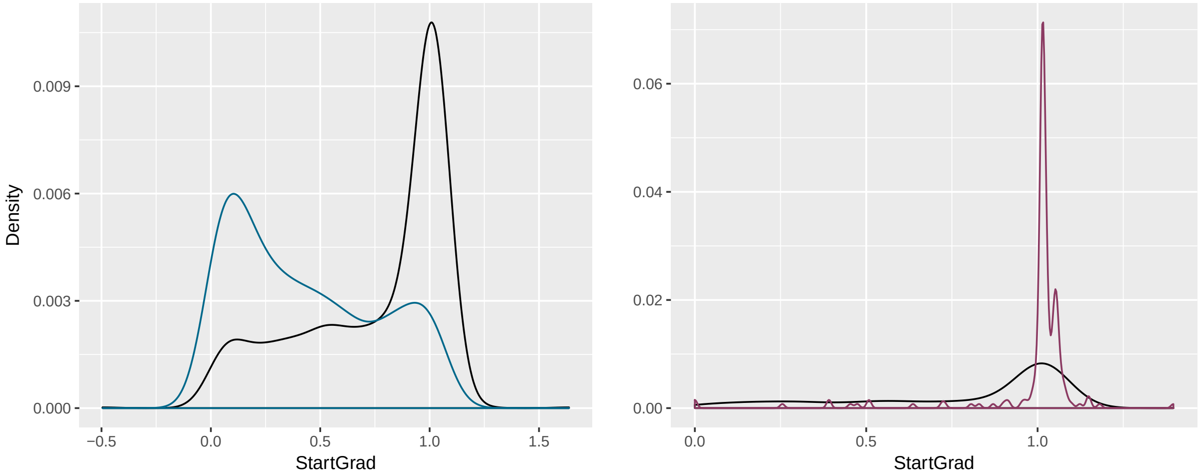


Figure 4.20: Comparison of the density of the covariate StartGrad in the support vectors (black) and the remaining correctly classified observations (blue, purple). The low-risk cases (RR $= 1$) is presented to the left, and the high-risk cases (RR $= 0$) is presented to the right.

We continue our comparison for the support vectors and the remaining observations by computing the average value of the quantitative variables in the dataset. Henceforth, the observations which are not set as support vectors, and therefore not influencing our model, are denoted the non-support vectors. From Table 4.9 it is observed that the average age of the accounts is highest for the non-support vectors with RR $= 1$, and lowest for the non-support vectors with RR $= 0$. This could imply that customers with high-risk are generally sent to collection earlier than low-risk customers. The result also suggests that older customers are in greater chance of recovering comparing to newer customers. However, we note that the difference in AgeOfCustomer is not prominent between the non-support vectors for the recovered accounts versus to the non-recovered accounts. The debt given in the application is considerably higher for the recovering accounts compared to the non-recovering accounts, independent of the observations being support vectors or not. The values for the support vectors are however, closer than for the non-support vectors. Either do high-risk accounts carry a generally lower debt, or they have the tendency to lie on their application. The bank's resources in investigating whether the application information is correct when granting a credit card to a new customer, is limited. There is therefore no reason for an applicant, who is in the need of a credit card, not to lie on their application. We also notice that the stated income in the application is higher for all accounts, which substantiates the previous statement regarding the integrity of the applicants. The amount of time before the credit card is used is higher for the recovering accounts. This supports the hypothesis of high-risk customers spending all their available credit balance immediately after receiving their credit card. The variable StartGrad also reveals that high-risk accounts spend more money from the first active month to the next.

Table 4.9: Average values for quantitative variables in the support vectors ($\mathrm{SV}_{\mathrm{RR}=1}$, $\mathrm{SV}_{\mathrm{RR}=0}$), and the remaining observations ($\overline{\mathrm{SV}}_{\mathrm{RR}=1}, \overline{\mathrm{SV}}_{\mathrm{RR}=0}$) for `data.new`

	$\overline{\mathrm{SV}}_{\mathrm{RR}=1}$	$\mathrm{SV}_{\mathrm{RR}=1}$	$\mathrm{SV}_{\mathrm{RR}=0}$	$\overline{\mathrm{SV}}_{\mathrm{RR}=0}$
SerialN	30.75	30.66	27.33	28.1
AgeOfAccount	4.48	4.44	4.16	4.12
AgeOfCustomer	35.96	35.16	34.11	34.79
MTPCollectionWarning	1639	1646	2505	2522
ClosingBalanceCollectionWarning	24812	22614	26373	27696
SumCollectionWarning	0.05	0.05	0.04	0.05
SumDunning	1.18	1.18	1.15	1.15
Recurring	0.02	0.01	0.01	0.01
SumCreditIncrease	0.03	0.02	0.01	0.01
DebtAppl	716661	645342	301837	257182
GrossIncomeAppl	557088	615381	376327	405899
GrossIncome	288414	279671	235121	237343
Active	3.98	3.96	3.78	3.77
RR	0.06	0.06	0.03	0.03
RR.avg	0.02	0.02	0.01	0.01
AvgClosingBalance	0.82	0.81	0.96	0.98
StartGrad	0.63	0.61	0.81	0.84
AvgNormal	0.42	0.42	0.41	0.40
AvgCashTransfer	0.19	0.18	0.23	0.25
AvgPurchase	0.08	0.09	0.07	0.06
AvgInterest	195.38	181.98	235.48	246.20
AvgPayN	0.21	0.19	0.15	0.15
AvgOverLimit	0.70	0.66	1.02	1.05
MaxClosingBalance	25051	22994	26482	277514
DaysSinceLastTime	9787	9863	9945	9894
CreditLimit	27067	25551	25475	26141
AvgCashWithdrawal	0.05	0.05	0.07	0.06
AvgLessThanMin	0.91	0.90	0.93	0.94
FirstBalance	15778	14460	20260	21726
FirstActive	1.50	1.46	1.37	1.34
Returned	0.03	−0.01	−0.04	−0.06
SumOverLimit	2.75	2.61	3.76	3.83
MaxCashTransfer	0.45	0.39	0.52	0.58
MaxCashWithdrawal	0.14	0.16	0.21	0.20

4.3.2 Model for Old Accounts

Model Selection

After observing an interaction between C and σ for the dataset containing the newer accounts, we instantly start with a CCD for the older account dataset. The initial values are set to $C = \{0.25, 4\}$ and $\sigma = \{4.35 \cdot 10^{-3}, 1.59 \cdot 10^{-2}\}$, in addition to the star points with a distance $\alpha = \sqrt{2}$ from the centre of the design space. The result is presented to the left in Figure 4.21, where we observe an optimal value of RSS $= 3.24 \cdot 10^{-5}$ in the point $(C = 4, \sigma = 4.35 \cdot 10^{-3})$. Continuing along the gradient from the centre of the design space to the first step optimum did not improve the value of RSS. we will, therefore, perform another CCD around this optimum. The second step optimal was calculated to be RSS $= 2.96 \cdot 10^{-5}$ in the point $(C = 3.5, \sigma = 2.35 \cdot 10^{-3})$ as indicated to the right in Figure 4.21. Continuing along the path from the centre of the design space to the second step optimum did not improve the prediction error. Hence, we terminate the optimization process stating we have found a local optimum.

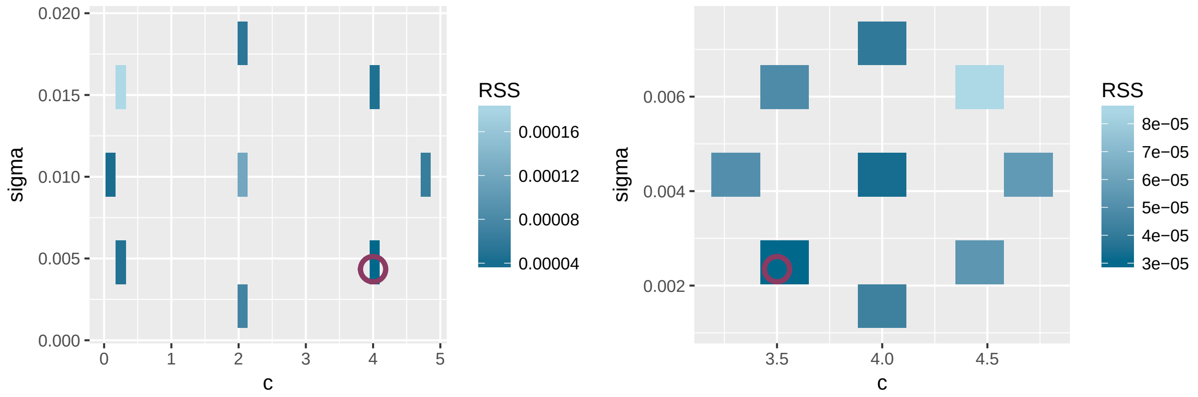


Figure 4.21: Optimization of the hyperparameters C and σ using the concept of RSM for `data.old`. The predictive error is presented in a heat map, and the purple circle denotes the optimal point of (C, σ) CCD in step 1 (left) and step 2 (right).

Predictive Performance

Fitting the SVM using the Gaussian RBF kernel with parameter values $C = 3.5$ and $\sigma = 2.35 \cdot 10^{-3}$ yields 11999 support vectors out of a total of 27581 observations. This again sheds some light on how inconsistent human behaviour can be, and is visualized in Figure 4.22. We notice a great overlap for predicted values between 0.75 and 0.85. The reason to why this overlap is not centreed at 0.5 as for `data.new`, is the more prominent imbalance in the data observed from Table 4.1b. In order to obtain a satisfactory predictive result we need to classify observations in the overlapping interval to the uncertain category.

The model's predictive performance is presented in Table 4.10, and reveals an accuracy $AC = 0.799$ and precision $P = 0.807$ when using the boundary $\hat{\pi} = 0.5$. However, we have already seen that this is not an optimal boundary. Including the uncertain category gives

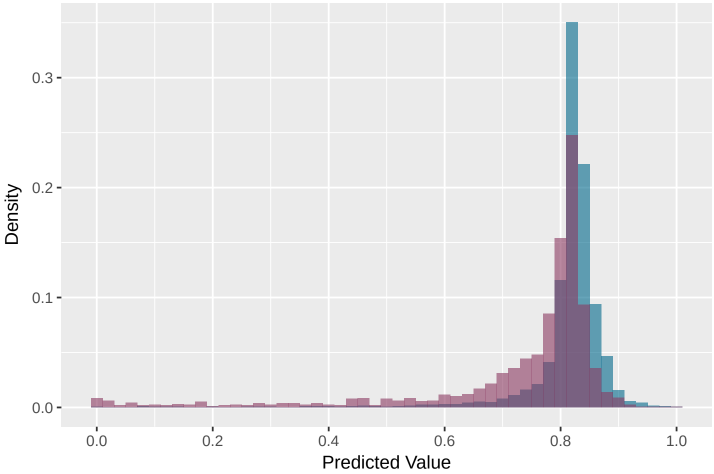


Figure 4.22: Distribution of the predicted values for both recovered (blue) and non-recovered (purple) accounts in the SVM model for `data.old`

the updated values $AC = 0.855$ and $P = 0.912$, which is close to the logistic model. It is worth mentioning that the SVM correctly specifies more negative cases compared to the logistic model, but unfortunately at the cost of incorrectly specifying positive cases. This is expected due to the high number of support vectors, which indicates that the SVM is not able to find a suitable hyperplane dividing the two categories. Considering the values for the accuracy and precision alone give no indication of an insufficient predictive power.

Table 4.10: Confusion matrices of the SVM for `data.old` with threshold $\hat{\pi} = 0.5$ (left), and $\hat{\pi} \in [0.75, 0.85]$ (right).

		Prediction	
		False	True
Reference	False	178	1719
	True	120	7177

		Prediction		
		False	F/T	True
Reference	False	178	1599	120
	True	120	5935	1242

Characteristics of the Support Vectors

A comparison between the support vectors and the remaining observations is done by looking at the average value for the quantitative variables in the dataset. Table 4.11 reveals that the average age of the customers is lowest for the non-support vectors with RR $= 1$, and highest for the non-support vectors with RR $= 0$, contrarily to the newer accounts. The assumption of older and more established customers having a higher probability of returning back to normal, is therefore not supported here. The average minimum payment required to recover is lower for the recovered accounts versus the non-recovered accounts. A low value of MTPCollection-Warning will therefore work as a incentive to recover. One interesting observation is the average number of times a dunning has been sent to the accounts, which is highest for the recovering accounts among the non-support vectors. In other words, the more frequently accounts are sent to collection and manage to recover, the more likely they are to recover later. The same interpretation yields for the average number of collection warnings and recurrences. We also notice that the non-recovering accounts have a higher rate of credit increase, and a higher gap between the income given in the application and the actual observed income for the same year.

Table 4.11: Average values for quantitative variables in the support vectors ($\mathrm{SV}_{\mathrm{RR}=1}$, $\mathrm{SV}_{\mathrm{RR}=0}$), and the remaining observations ($\overline{\mathrm{SV}}_{\mathrm{RR}=1}, \overline{\mathrm{SV}}_{\mathrm{RR}=0}$) for `data.old`

	$\overline{\mathrm{SV}}_{\mathrm{RR}=1}$	$\mathrm{SV}_{\mathrm{RR}=1}$	$\mathrm{SV}_{\mathrm{RR}=0}$	$\overline{\mathrm{SV}}_{\mathrm{RR}=0}$
SerialN	31.62	31.52	29.29	28.99
AgeOfAccount	55.86	56.43	47.05	45.70
AgeOfCustomer	39.18	39.23	39.32	39.45
MTPCollectionWarning	1860	1851	2926	2984
ClosingBalanceCollectionWarning	28937	27869	34871	35848
SumCollectionWarning	1.94	1.71	1.30	1.43
SumDunning	5.38	4.84	3.88	4.15
Recurring	1.08	0.93	0.74	0.86
SumCreditIncrease	0.27	0.22	0.39	0.42
DebtAppl	385226	490036	152541	821986
GrossIncomeAppl	188628	173773	184063	263321
GrossIncome	120448	100341	118516	136043
Active	11.37	11.36	11.09	11.04
RR	0.10	0.11	0.08	0.08
RR.avg	0.09	0.09	0.07	0.08
AvgClosingBalance	0.82	0.81	0.90	0.90
AvgNormal	0.54	0.54	0.57	0.57
AvgCashTransfer	0.05	0.05	0.07	0.07
AvgPurchase	0.04	0.04	0.03	0.03
AvgInterest	428.38	411.91	530.83	559.78
AvgPayN	0.48	0.47	0.50	0.51
AvgOverLimit	0.54	0.52	0.71	0.72
MaxClosingBalance	30682	29825	36235	37003
CreditLimit	30793	30336	33290	34143
AvgCashWithdrawal	0.02	0.02	0.03	0.02
AvgLessThanMin	0.95	0.95	1.03	1.03
MaxCashTransfer	0.19	0.18	0.25	0.26
MaxCashWithdrawal	0.09	0.10	0.14	0.13

4.4 Fuzzy Clustering Analysis

Intuitively, one could imagine that different trends in behaviour leads to the same outcome. In other words, it could be interesting to investigate whether accounts could be grouped into more than two clusters. This is the main aim of including fuzzy clustering in this analysis. As presented in section 3.3, there are several fuzzy algorithms to choose from. we will, therefore, start with FCM with norm-inducing matrix **A** = **I** as a base case, before exploring GK and PFCM.

4.4.1 Model for New Accounts

To get a preliminary insight into the cluster structure of the data, we will first look at a visual cluster assessing of the data. The blue cells in Figure 4.23 indicate low dissimilarity, while yellow cells indicate high dissimilarity. The plot reorganizes the dissimilarities such that diagonal blocks correspond to clusters in the data. We notice that the potential number of clusters could be $K = 3$ or even $K = 5$.

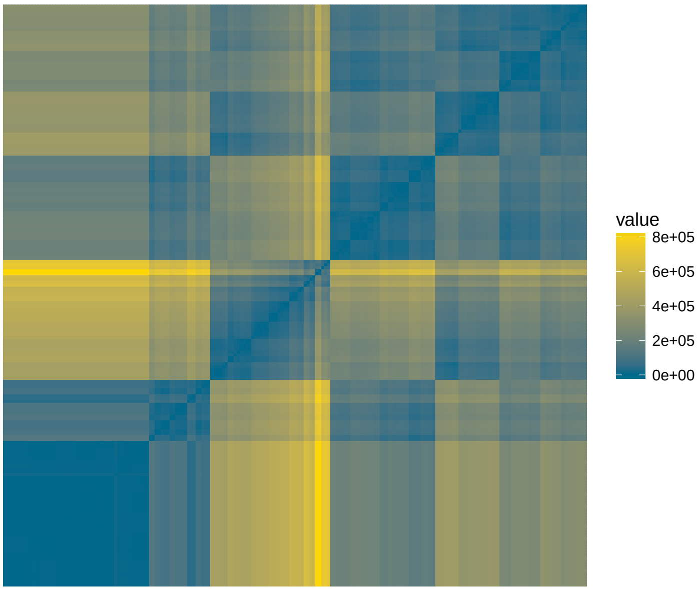


Figure 4.23: Visual assessing of cluster tendency for `data.new`

Fuzzy c-means (FCM)

As with SVM, the FCM algorithm includes two hyperparameters K and m. In order to find the optimal value of these two, a grid search is performed using the validity measures V_{XB} and V_{FS} presented in (3.49) and (3.50), respectively. For K, maximum value is 10, minimum value is 2 and step length is 1. For m, maximum value is 2.5, minimum value is 1.5 and step length is 0.25. Then, models using all the grid points are fitted in order to detect which one yields the optimal validity measures. Section 3.3 call attention to the problems occurring when the dimensions of the data become too high, and we will, therefore, fit the FCM including the preserved features from applying the elastic net with $\alpha = 0.7$. We use the function cmeans in the package **e1071** for fitting the FCM. Figure 4.24 is the heat map of the Xie-Beni and Fukuyama Sugeno index using the k-means++ and al-daoud as prototype initialization on data.new. We notice that the two initializations produce consistent results, with an optimal number of clusters $K = 3$ confirming our assumptions from Figure 4.23. The optimal fuzziness is calculated to be $m = 2.25$. This indicates that there does not exist that many different behavioural patterns, but dividing the data into three clusters is the optimal choice.

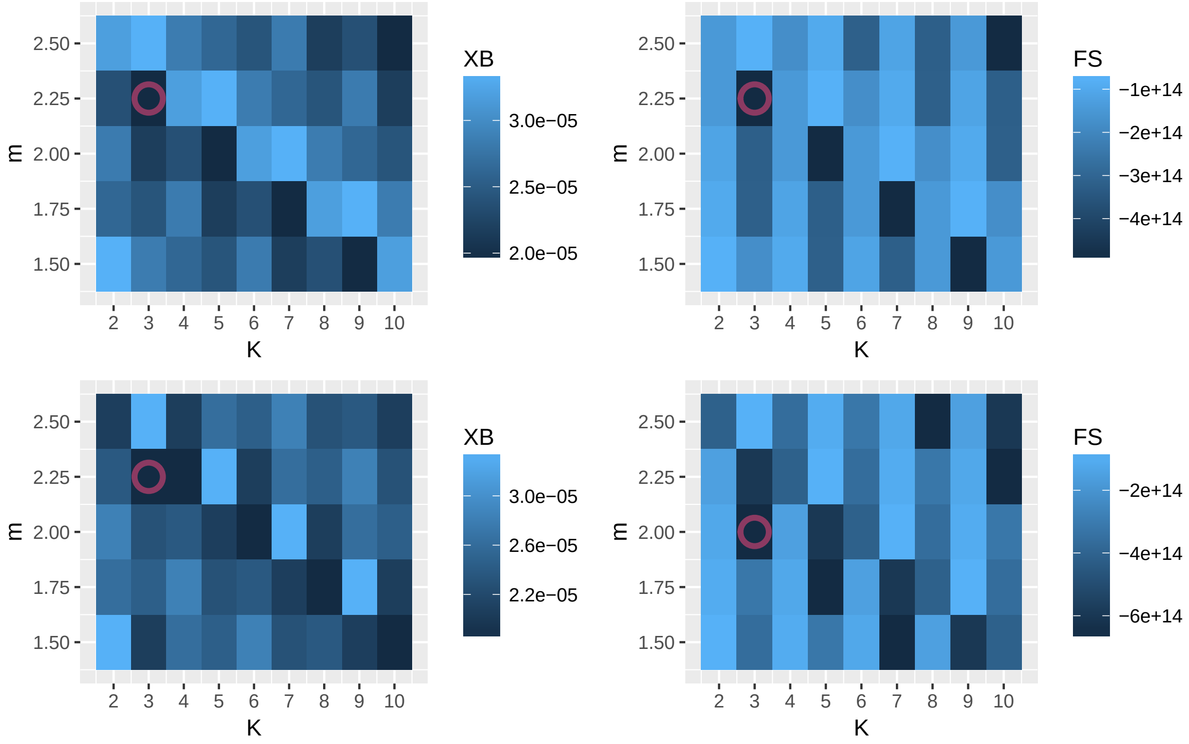


Figure 4.24: Grid search of the hyperparameters K and m for data.new. The Xie-Beni index with k-means++ prototype initialization (top left), the Fukuyama Sugeno index with k-means++ prototype initialization (top right), the Xie-Beni index with Al-Daoud prototype initialization (bottom left), and the Fukuyama Sugeno index with Al-Daoud prototype initialization (bottom right).

Fitting a FCM model with parameters $K = 3$ and $m = 2.25$ yields the results given in Table 4.12. When assigning the observations to the closest hard clustering, i.e the class with maximal membership, we observe that cluster 3 has an overweight of zero (RR $= 0$) cases, while cluster 2 has an overweight of positive (RR $= 1$) cases. However, the overweight in cluster 3 is only 55%, which is not high enough to label the cluster. Instead of assigning the observations to the closest hard cluster, we could take advantage of the fuzziness of the predictions by only assigning an observation to a cluster if its membership is higher than some threshold. However, when looking at the distribution of maximum membership values in Figure 4.25 nearly all of them are found to be $\approx 1/3$. This is also confirmed from the partition coefficient calculated to

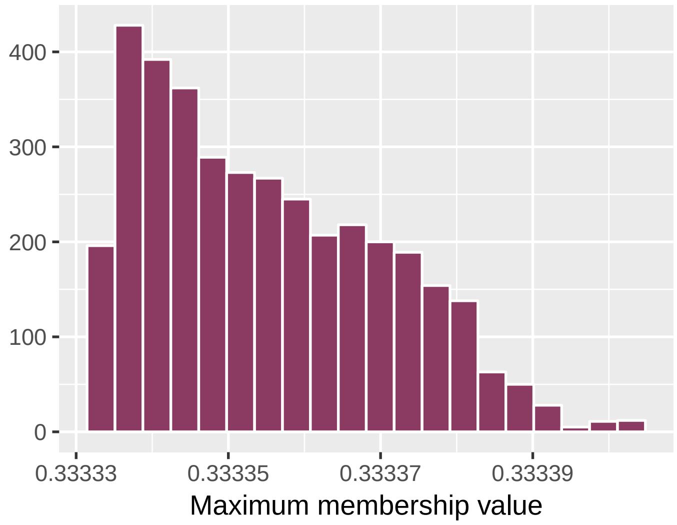


Figure 4.25: Distribution of the maximal membership values for the FCM with $K = 3$ and $m = 2.25$ for `data.new`

be $V_{PC} = 1/3$. This means that the cluster assignments presented in Table 4.12 are just a result of small differences in the observations, and that essentially all observations are equally typical for all clusters.

Table 4.12: Cluster results when assigning observations to the cluster of maximal membership using the FCM on `data.new`

		Cluster		
		1	2	3
Reference	0	42	228	836
	1	33	972	684

Gustafson-Kessel (GK)

As the method of FCM did not prove to produce a satisfactory result, we continue with the Gustafson-Kessel method which allows for the clusters to be of different geometrical shapes. Letting the fuzziness parameter still have the value $m = 2.25$, we fit several models with different number of clusters ranging from 2 to 9 using the gk function from the **ppclust** package. The reason for this approach is to test the hypothesis of the existence of several behavioural trends leading to an equal outcome, instead of only two or three clusters. Table 4.14a presents the cluster assignments for each of the eight models. Assigning observations to the cluster with highest membership value results in several empty clusters, which is an unfortunate property. However, analyzing all of them simultaneously may shed some light on distinct features of recovering and non-recovering accounts.

For instance, when dividing the dataset into $K = 9$ clusters we notice that cluster number 4 include 88.1% recovering accounts. From the data we observe that this cluster has the lowest prototype value among all the clusters for the covariate Returned. From Table A.2 in appendix A we have that a low number indicates customers who once have paid their invoices are no longer able or willing to pay. Intuitively, one would expect the opposite result, but further investigation shows that among the observations assigned to cluster 4 only 1 among all 13 observations which had Returned $= -1$ did not recover. Hence, a value of Returned $= -1$ is not an implication of high-risk for customers in this cluster. Another observation made is that this cluster has the second highest prototype value for the covariate FirstActive, which means that the observations in this cluster spend more than the average amount of time before they start using their credit card. The mean value of FirstActive is 1.43 for all customers and 2.10 for customers in cluster 4. Among the customers in cluster 4 with FirstActive > 2, only 2 out of 16 observations did not recover. It is also noticed that among the customers in cluster 4 which have values Returned $= -1$ and FirstActive > 2, nearly all which have EvaluationMethodOrdinary $= 0$ did recover. It is therefore interesting to investigate if this combination yields the same result for all observations, and not just within this particular cluster. It is found that among all observations which have Returned $= -1$, FirstActive > 2 and EvaluationMethod $\neq$ Ordinary, will 88.9% recover.

Although cluster number 1 only includes two observations, both of them are recovering accounts, and so we want to examine what is common for these two observations. We notice that both of them have AvgInterest < 25, MaxCashWithdrawal ≤ 0.05 and EvaluationMethodOrdnary $= 0$. Testing this hypothesis for all the observations yields a share of 91.6% of recovering accounts. So, despite the fact that our cluster assignments were not optimal when dividing into $K = 9$ clusters, we were still able to detect some trends which lead to an outcome of RR $= 1$.

The majority of the observed clusters have a heavier weight of recovering accounts, which is not surprising given the unbalanced data observed from Table 4.1a. Furthermore, for the clusters which have an overweight of non-recovering accounts, we observe that this overweight is not dominating. The highest fraction of non-recovering accounts in a cluster is calculated to be 62.3%, observed when $K = 4$. This particular cluster has the highest prototype values for the features AvgOverLimit, AvgLessThanMin, MaxCashWithdrawal, AvgClosingBalance, SerialN10 and ProductSpareBank 1 MasterCard Gold. The mean value of AvgOverLimit is 0.818 for all customers and 0.999 for customers in cluster 4. Among the customers in cluster 4 with AvgOverLimit > 1 are there 74.1% who did not recover. Similarly, the mean value of AvgLessThanMin is 0.917 for all customers and 0.939 for customers in cluster 4. The fraction of non-recovering accounts among those who have AvgOverLimit > 1 and AvgLessThanMin > 0.94 is 76.1%. Testing this hypothesis on the whole dataset, and not just for this particular cluster, yields a share of 63.1% of non-recovering accounts with these feature values. It is, however, a major drawback that the algorithm itself could not draw these conclusions. we will, therefore, continue with the possibilistic fuzzy c-means model using the function pfcm in the **ppclust** package.

Possibilistic Fuzzy c-means (PFCM)

In order to decide which values of a, b, m and η we should use for the PFCM model, we look at the validity indices V_{PC}, V_{PE} and V_{XB} for different values of the tuning parameters. The number of clusters is assigned to be $K = 3$ since our goal is to divide the accounts into three groups, in addition to the observations made from Figure 4.23. We fit three linear models assuming that each validity index can be expressed as

$$\begin{aligned}
V_{PE} &= \beta_0 + \beta_a a + \beta_b b + \beta_m m + \beta_\eta \eta + \beta_{a,b} a \cdot b + \beta_{a,m} a \cdot m + \beta_{b,m} b \cdot m + \varepsilon \\
V_{PC} &= \tilde{\beta}_0 + \tilde{\beta}_a a + \tilde{\beta}_b b + \tilde{\beta}_m m + \tilde{\beta}_\eta \eta + \tilde{\beta}_{a,b} a \cdot b + \tilde{\beta}_{a,m} a \cdot m + \tilde{\beta}_{b,m} b \cdot m + \varepsilon \\
V_{XB} &= \beta_0^* + \beta_a^* a + \beta_b^* b + \beta_m^* m + \beta_\eta^* \eta + \beta_{a,b}^* a \cdot b + \beta_{a,m}^* a \cdot m + \beta_{b,m}^* b \cdot m + \varepsilon.
\end{aligned}$$

The estimates are found by using a 2^{4-1} factorial design, where the respective low and high values for the hyperparameters are set to be $a = \{1, 2\}, b = \{1, 5\}, m = \{2, 5\}$ and $\eta = \{1.5, 3\}$. The sign matrix together with the corresponding response values are presented in Table 4.13, and the estimates are given in appendix B.

Table 4.13: Sign matrix with level codes for a, b, m and $\eta = abm$, and respective validity indices when $K = 3$ for `data.new`

Run	a	b	m	$\eta = abm$	V_{PE}	V_{PC}	V_{XB}
1	$-$	$-$	$-$	$-$	0.384	0.978	76468
2	$-$	$-$	$+$	$+$	0.939	0.508	84603
3	$-$	$+$	$-$	$+$	0.783	0.358	50693
4	$-$	$+$	$+$	$-$	0.201	0.565	162650
5	$+$	$-$	$-$	$+$	0.919	0.519	617638
6	$+$	$-$	$+$	$-$	0.390	0.924	147842
7	$+$	$+$	$-$	$-$	0.194	0.533	96561
8	$+$	$+$	$+$	$+$	0.825	0.229	114096

Figure B.2 in appendix B reveal that only the value of b and η have significant effects on the partition entropy and partition coefficient. For the partition entropy, the effects are negative for b and positive for η, while both negative for the partition coefficient. The property of a decreased partition entropy when the value of b is increased implies an existence of outliers, which causes for our clusters not to be well separated. Hence, increasing the value of b reduces the effect of outliers and improves the value of V_{PE}. We notice that the Xie-Beni index is independent of all the parameters, and so our decision will solely be based on the partition entropy and partition coefficient. A low value of η makes the typicality assignments crisp, and since the cluster assignments for the PFCM are given by the typicality values, the fuzzy solution will become closer to the hard solution. This will increase the value of the partition coefficient. Simultaneously, a high value of η yields a high partition entropy which means that our clusters are becoming more overlapped. An acceptable trade-off between the partition entropy and partition coefficient is observed for run 1 where all the coefficients are at its low level.

Opposed to the Gustafson-Kessel model, Table 4.14b reveals that observations are assigned to all of the PFCM clusters. However, it does not seem that the groupings can be characterized by the RR as nearly all of the clusters obtains a heavier weight of recovering accounts. This could imply that the clustering problems are not rooted in the dimensionality of the data, but rather the lack of groupings with respect to the recovery rate.

Table 4.14: Cluster results from applying the Gustafson-Kessel **(a)** and the Possibilistic fuzzy c-means **(b)** for values of $K = 2, ..., 9$, and defuzzifying the membership degrees (GK) and typicality degrees (PFCM) of the objects for `data.new`.

(a)

K	Cluster	Prediction	
		0	1
2	1	114	67
	2	992	1622
3	1	71	47
	2	1035	1642
	3	0	0
4	1	997	1623
	2	0	0
	3	0	0
	4	109	66
5	1	66	40
	2	101	141
	3	0	0
	4	65	66
	5	874	1442
6	1	12	8
	2	0	0
	3	0	0
	4	233	240
	5	0	0
	6	861	1441
7	1	106	303
	2	6	4
	3	897	1322
	4	0	0
	5	73	47
	6	24	13
	7	0	0
8	1	24	12
	2	0	0
	3	7	5
	4	0	0
	5	730	1100
	6	269	525
	7	0	0
	8	76	47
9	1	0	2
	2	79	48
	3	0	0
	4	7	52
	5	686	778
	6	0	0
	7	95	269
	8	239	540
	9	0	0

(b)

K	Cluster	Prediction	
		0	1
2	1	519	923
	2	587	766
3	1	182	414
	2	377	520
	3	547	755
4	1	203	347
	2	332	434
	3	456	635
	4	115	273
5	1	128	196
	2	323	401
	3	444	651
	4	103	202
	5	108	239
6	1	111	205
	2	316	395
	3	181	249
	4	70	156
	5	20	61
	6	408	623
7	1	25	37
	2	313	392
	3	205	284
	4	103	196
	5	18	58
	6	376	572
	7	66	150
8	1	23	47
	2	312	392
	3	20	67
	4	102	182
	5	66	156
	6	181	261
	7	88	176
	8	314	408
9	1	123	181
	2	320	397
	3	70	157
	4	97	90
	5	92	199
	6	274	405
	7	94	144
	8	4	17
	9	32	99

4.4.2 Model for Old Accounts

To obtain some notion of the cluster structure, we look at the ordered dissimilarity plot presented in Figure 4.26. The cluster structure is not as visible as for the newer dataset, but an optimal value of $K = 2$ seems plausible.

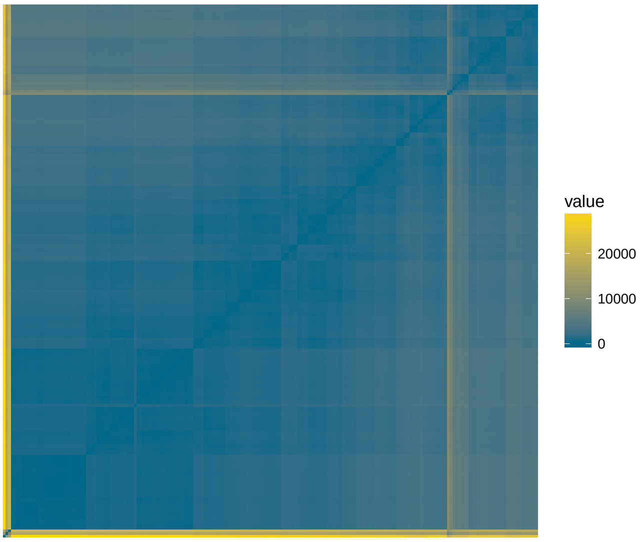


Figure 4.26: Visual assessing of cluster tendency for `data.old`

We will here perform the same analysis as for the dataset containing the newer accounts by first looking at the FCM algorithm before continuing with the GK and finally the PFCM.

Fuzzy c-means (FCM)

Similarly to the newer account dataset, we perform a grid search for the values of m and K using the validity indices V_{XB} and X_{FS} as presented in Figure 4.27. The optimal parameter values differ somewhat between the two different validity indices but are consistent with the different initialization techniques. we will, therefore, continue with fitting two FCM models using the optimal values indicated from the two measures. We start with $K = 2$ and $m = 2.5$ as indicated from the Xie-Beni index, and then continue with $K = 3$ and $m = 1.75$ from the Fukuyama-Sugeno index. Fitting the first FCM model including the preserved covariates from performing the elastic net regularization on the data with $\alpha = 0.9$, yields the result presented in Table 4.15.

Table 4.15: Cluster results when assigning observations to the cluster of maximal membership (left) and when membership $u_{nk} > 0.9$ (right) using the FCM with $K = 2$ and $m = 2.5$ on `data.old`

		Cluster	
		1	2
Reference	0	5306	2182
	1	25156	4131

		Cluster		
		1	2	0.5
Reference	0	3395	474	3619
	1	19116	849	9322

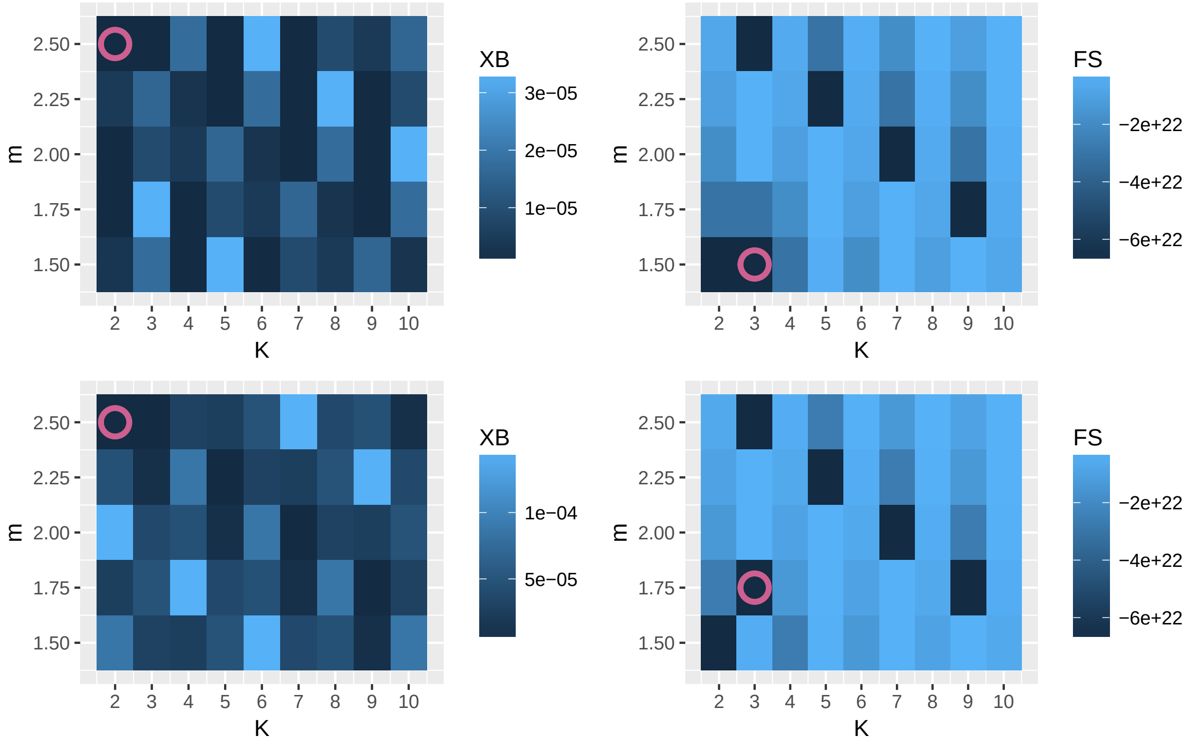


Figure 4.27: Grid search of the hyperparameters K and m for `data.old`. The Xie-Beni index with k-means++ prototype initialization (top left), the Fukuyama Sugeno index with k-means++ prototype initialization (top right), the Xie-Beni index with Al-Daoud prototype initialization (bottom left), and the Fukuyama Sugeno index with Al-Daoud prototype initialization (bottom right).

When assigning the observations to the closest hard clustering both clusters are dominated by recovering accounts. By only assigning observations to the closest hard clustering if the corresponding membership value is $u_{nk} > 0.9$, the ratio between non-recovering and recovering accounts is preserved. we will, therefore, fit the second FCM model with result presented in Table 4.16. Note that cluster 0.5 corresponds to observations which have a maximal membership lower than 0.9, and are therefore not assigned to either of the clusters. Just as for $K = 2$, we observe only clusters dominated by recovering accounts.

Table 4.16: Cluster results when assigning observations to the cluster of maximal membership (left) and when membership $u_{nk} > 0.9$ (right) using the FCM with $K = 3$ and $m = 1.75$ on `data.old`

		Cluster		
		1	2	3
Reference	0	4493	2650	345
	1	22883	6013	391

		Cluster			
		1	2	3	0.5
Reference	0	3671	1509	150	2158
	1	20029	3260	167	5831

Gustafson-Kessel (GK)

There is no indication that choosing $K = 3$ improves the partition of non-recovering and recovering accounts when looking at both the closest hard clustering and when the membership values exceed a threshold of $u_{nk} > 0.9$. Just as with the newer accounts dataset, we will, therefore, proceed with the Gustafson-Kessel algorithm letting $K = 2, ..., 9$ while keeping $m = 1.75$ constant. Table 4.18a gives the cluster assignments for each of the eight models, where we notice that we are still not achieving a satisfactory partitioning of accounts based on their recovery rate. In fact, we have no cluster for which non-recovering accounts are the majority. The reason could simply be the high dimension of `data.old` compared to `data.new`, but dividing the dataset into smaller subsets and fitting the GK algorithm on a subset of equal size to `data.new` produces similar results. Considering Figure 4.26 and the fact that several clusters were not assigned any observations, it is not that surprising that our data do not exhibit a clustering structure. Actually, there is no value of K where the algorithm successfully detects the given number of clusters. Some analysis could be performed on the clusters having an overweight of recovering accounts, similar to what was done for `data.new`. The reason why this is not done here, is simply because the overweight is not higher for each cluster compared to the whole dataset.

Possibilistic Fuzzy c-means (PFCM)

As a final attempt, we will apply the PFCM model and use the partition coefficient, partition entropy and Xie-Beni index to choose between different values for the parameters a, b, m and η. Due to the poor result of the GK algorithm when applying high values of K, and the implication of $K = 2$ from Figure 4.26, we let $K = 2$ when performing the PFCM. Similar to the hyperparameters in the SVM, we assume that each validity index can be expressed as

$$\begin{aligned}
V_{PE} &= \beta_0 + \beta_a a + \beta_b b + \beta_m m + \beta_\eta \eta + \beta_{a,b} a \cdot b + \beta_{a,m} a \cdot m + \beta_{b,m} b \cdot m + \varepsilon \\
V_{PC} &= \tilde{\beta}_0 + \tilde{\beta}_a a + \tilde{\beta}_b b + \tilde{\beta}_m m + \tilde{\beta}_\eta \eta + \tilde{\beta}_{a,b} a \cdot b + \tilde{\beta}_{a,m} a \cdot m + \tilde{\beta}_{b,m} b \cdot m + \varepsilon \\
V_{XB} &= \beta^*_0 + \beta^*_a a + \beta^*_b b + \beta^*_m m + \beta^*_\eta \eta + \beta^*_{a,b} a \cdot b + \beta^*_{a,m} a \cdot m + \beta^*_{b,m} b \cdot m + \varepsilon.
\end{aligned}$$

Using a 2^{4-1} factorial design by letting the respective low and high values of the hyperparameters be set to $a = \{1, 2\}, b = \{1, 5\}, m = \{2, 5\}$ and $\eta = \{1.5, 3\}$, we obtain the validity indices reported in Table 4.17. Section 3.3.5 states that the partition coefficient should not take values $V_{PC} > 1$. This is due to the possibilistic constraint on the membership values, which the calculation of this index is based on. For the PFCM is V_{PC} calculated from the typicality values which are not constrained to sum up to 1, and so it can take on values higher than 1. However, a value higher than 1 is not desired as it implies that observations attain high typicality for several clusters. In other words, it is difficult deciding the optimal parameter values based on the partition coefficient alone.

The estimates and corresponding half-normal plots in Figure B.3 are presented in appendix B, and reveal that η has a significant positive effect on the partition entropy and a significant negative effect on the partition coefficient. The value of b has a significant negative effect on the partition coefficient, while the interaction of a and m has a significant positive effect on the

Table 4.17: Sign matrix with level codes for a, b, m and $\eta = abm$, and respective validity indices when $K = 2$ for `data.old`

Run	a	b	m	$\eta = abm$	V_{PE}	V_{PC}	V_{XB}
1	$-$	$-$	$-$	$-$	0.124	1.598	1639330
2	$-$	$-$	$+$	$+$	0.531	0.837	1335348
3	$-$	$+$	$-$	$+$	0.525	0.458	162332
4	$-$	$+$	$+$	$-$	0.141	0.769	183322
5	$+$	$-$	$-$	$+$	0.554	0.779	288080
6	$+$	$-$	$+$	$-$	0.134	1.591	2243384
7	$+$	$+$	$-$	$-$	0.116	0.744	181313
8	$+$	$+$	$+$	$+$	0.638	0.466	808281

partition coefficient. The model fit on the Xie-Beni index revealed no significant effects, and so we will focus on the two other indices when deciding the hyperparameter values. The positive effect on the partition entropy tells us that increasing η causes for overlapping clusters. Increasing a and m simultaneously results in the membership component of the objective function in (3.44) trying to push the prototypes towards the grand mean, making typicality more important for centroid computation. This is consistent with the negative estimate of η as decreasing this parameter makes typicality assignments almost crisp causing for the fuzzy solution to become closer to the hard solution. Since the partition coefficient is calculated from the typicality values for the PFCM, this will increase the value of V_{PC}. We will, therefore, seek a small value of η, while high for a and m. Increasing b from 1 to 5 while maintaining the other parameter values fixed, causes for the partition coefficient to decrease, and so we want to avoid a low value of b. However, since we aim to keep $V_{PC} \leq 1$, the values of b and $a \cdot m$ cannot be at their respective high levels simultaneously. Both run 4 and 7 produce a satisfactory combination of the validity indices, and so we will fit the PFCM using the values from run 4.

Figure 4.18b reveals that for each value of K there is one cluster which size is much larger than for the other clusters, just as for `data.new`. Opposed to the newer dataset, we do not obtain any cluster where the 0-cases dominate, but we do observe some clusters with dominating 1-cases. For instance, cluster 4 when $K = 9$ and cluster 4 when $K = 6$ have a fraction of 93.5% and 89.3% recovering cases, respectively. The final prototype values for both clusters reveal the same behaviour. Both clusters are centred at low values of MTPCollectionWarning, SumDunning, SumCreditIncrease, AvgInterest and AvgOverLimit compared to the other cluster centres. In other words, low-risk accounts typically have a lower amount they have to pay in order to recover, which indicate that the balance sent to collection is far less compared to high-risk accounts. The low value of SumDunning contradicts the findings from both the logistic - and SVM model, but could indicate the existence of some interaction between the number of payment reminders and the other variables. The observations within these clusters have increased the credit limit far less than the other observations, in addition to a lower incurred interest and number of overdrafts. Investigating these findings quantitatively by looking at the observations which fulfills MTPCollectionWarning < 350, SumDunning ≤ 4, SumCreditIncrease ≤ 1, AvgInterest < 150, AvgOverLimit < 0.2, yields a fraction of $2179/2388 = 91.2\%$ recovering accounts when looking at the whole dataset. Hence, we are able to seclude about 10% of the observations, meaning that we do not have to worry about whether these observations will recover. We discover that it is far easier to detect recovering cases than the opposite.

Table 4.18: Cluster results from applying the Gustafson-Kessel **(a)** and the Possibilistic fuzzy c-means **(b)** for values of $K = 2, ..., 9$, and defuzzifying the membership degrees (GK) and typicality degrees (PFCM) of the objects for `data.old`.

(a)

K	Cluster	Prediction	
		0	1
2	1	7488	29287
	2	0	0
3	1	7079	28593
	2	0	0
	3	409	694
4	1	0	0
	2	6398	24921
	3	681	3672
	4	409	694
5	1	0	0
	2	0	0
	3	918	4614
	4	0	0
	5	6570	24673
6	1	0	0
	2	0	0
	3	406	689
	4	7079	28594
	5	0	0
	6	3	5
7	1	0	0
	2	0	0
	3	125	295
	4	374	642
	5	0	0
	6	6965	28306
	7	24	44
8	1	0	0
	2	349	630
	3	0	0
	4	0	0
	5	0	0
	6	0	0
	7	7139	28657
	8	0	0
9	1	0	0
	2	0	0
	3	0	0
	4	0	0
	5	0	0
	6	0	0
	7	6441	26099
	8	440	754
	9	607	2434

(b)

K	Cluster	Prediction	
		0	1
2	1	1389	2817
	2	6099	26470
3	1	59	89
	2	182	1019
	3	7247	28179
4	1	92	158
	2	239	1179
	3	7099	27618
	4	158	332
5	1	63	96
	2	169	882
	3	319	1399
	4	55	347
	5	6882	26463
6	1	47	102
	2	73	71
	3	274	1216
	4	44	368
	5	6756	26022
	6	294	1508
7	1	46	279
	2	279	1386
	3	338	1463
	4	37	360
	5	6350	24030
	6	389	1902
	7	49	47
8	1	30	66
	2	37	41
	3	406	1763
	4	24	334
	5	6173	23264
	6	438	2043
	7	51	101
	8	329	1675
9	1	32	73
	2	40	42
	3	196	797
	4	27	391
	5	5878	21984
	6	360	1885
	7	59	111
	8	317	1691
	9	579	2313

4.5 Sensitivity Analysis by Design of Experiments

One concern when building predictive models is their time of relevance. How much time needs to pass before these models no longer apply? It is therefore important to consider external factors which most likely will change with time, and evaluate the effect it will have on the model's predictive power. Opposed to the previously mentioned statistical models where we have performed observational studies, i.e we do not have any influence on the variables we are measuring, Design of Experiments (DoE) allows us to control the covariate values X of the process and then measure the response Y. The intention is to discover which values of the independent variables allow for improvement of the performance of our model, and which values causes for our model to perform poorly. By examining the variables included in the models built in the previous sections, we see that there are several predictors that might change severely in the future.

According to a newly published article by the Business Insider Nordic [44], less than 10% of all transactions made in Norway are in cash. Jon Nicolaisen, the deputy governor of Norway's central bank, argued at the City Week conference at London's Guildhall this year that Norway could be considered a cashless country. There is, therefore, reason to believe that all cash transactions will be equal to zero at some point in the future. The interest rate is known to change. If it will increase or decrease in the near future is uncertain, and it is therefore interesting to measure its impact. Today, the interest rate is about 25% depending on the amount of outstanding credit balance, if the invoice is issued by mail or electronically, etc. One other variable which often increases with time is the amount of income due to inflation or seniority.

Our sensitivity analysis will concern the models obtained using the method of support vector machine. Our parameters of interest will be AvgInterest, MaxCashWithdrawal/AvgCashWithdrawal and GrossIncome/GrossIncomeAppl when performing a 2^k factorial design with $k = 3$. It is important to stress that the analysis is done by changing the values of these variables in our test set, and further assume that our response remains constant when performing prediction on our fitted model. This might not conform to reality as a change in external factors can drive us to act differently.

We want to examine the effect on the accuracy and precision of the model. The low (-1) and high $(+1)$ levels for the given covariates are defined as

$$\begin{aligned} \text{AvgInterest (A)}: &\quad \text{interest rate equal to } 20\%(-1) \text{ or } 30\%(+1) \\ \text{CashWithdrawal (B)}: &\quad 0\ (-1) \text{ or today's value } (+1) \\ \text{Income (C)}: &\quad \text{today's value } (-1) \text{ or } 10\%(+1), \end{aligned} \tag{4.2}$$

and the results on both datasets are given in Table 4.19. We can already observe that neither the accuracy nor precision are highly influenced by these covariates in either of the datasets. Despite both the accuracy and the precision only taking values in the range $[0, 1]$ which motivates for beta regression, we will approximate the effects through the linear approach as the range of change in both measures are relatively small.

Table 4.19: Sign matrix with level codes and response values for `data.new` **(a)** and `data.old` **(b)**

(a)

A	B	C	AC	P
+	+	+	0.6888	0.7048
+	+	−	0.6899	0.7072
+	−	+	0.6899	0.6971
+	−	−	0.6899	0.6989
−	+	+	0.6877	0.6950
−	+	−	0.6856	0.6953
−	−	+	0.6835	0.6831
−	−	−	0.6813	0.6827

(b)

A	B	C	AC	P
+	+	+	0.7994	0.8069
+	+	−	0.7999	0.8073
+	−	+	0.7989	0.8042
+	−	−	0.7992	0.8045
−	+	+	0.7989	0.8047
−	+	−	0.7992	0.8050
−	−	+	0.7991	0.8024
−	−	−	0.7989	0.8025

We fit two linear models with AvgInterest, CashWithdrawal and Income as covariates, and the accuracy and precision as respective response values. The estimated effects for each dataset are given in Figure 4.28 and 4.30.

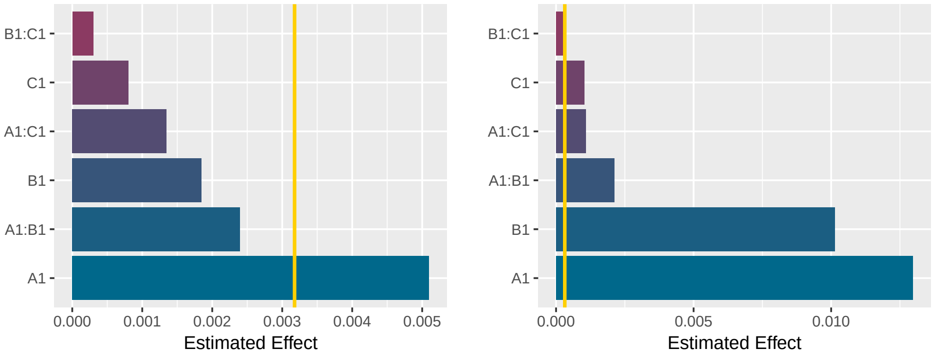


Figure 4.28: Estimated effects of accuracy (left) and precision (right) together with the line of significance for `data.new`. The definitions of the covariates A, B and C are given in (4.2).

Figure 4.28 reveals that the only covariate which has a significant effect on the accuracy for the newer dataset is the average incurred interest. Contrarily, we observe that the precision is significantly dependent on all the covariates and interactions. Hence, the accuracy of the model is more robust than the precision.

Examining the significant effects further we see that decreasing the interest rate causes a disadvantageous influence on both the accuracy and the precision. Decreasing the amount of withdrawn cash reduces the precision rate, which can be expected as the model is fitted using today's values. However, we notice that the value of cash withdrawals cannot decrease below 0, and so the covariate will not contribute to a lower precision than already observed in Table 4.28. The impact of increased income is not as dominant as for the other covariates, but still significant for the precision. One can, therefore, expect that as income increases, the model's precision rate will also continue to decrease.

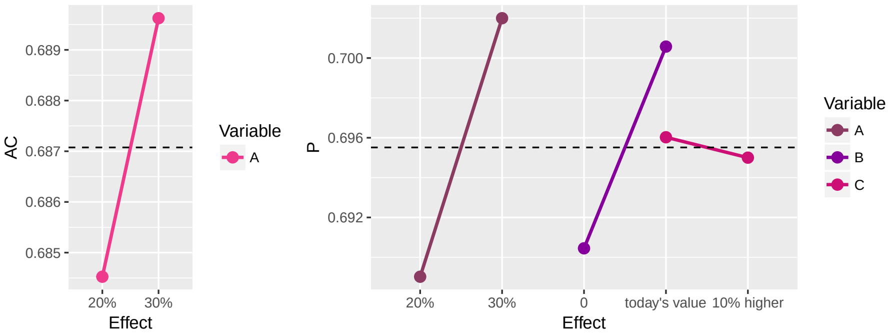


Figure 4.29: Significant main effects on accuracy (left) and precision (right) for `data.new`

We obtain no significant effects on the accuracy for the older dataset, implying that the accuracy will remain unchanged despite the changes given in (4.2). For the precision, on the other hand, we achieve significant effects for the interest rate and cash withdrawals. As for the older accounts, is the precision more vulnerable to change compared to the accuracy.

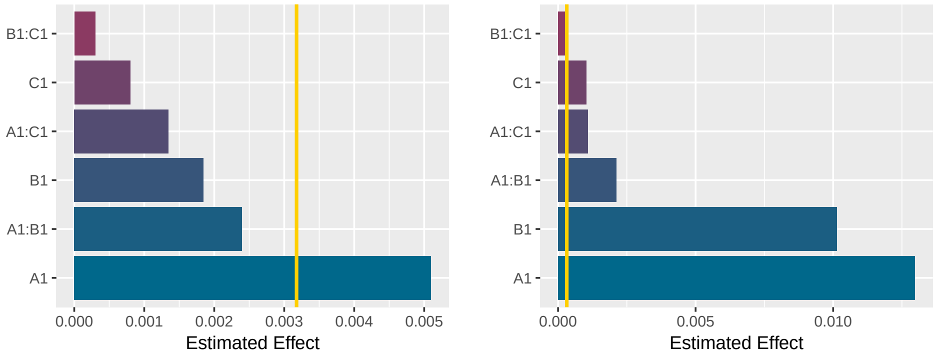


Figure 4.30: Estimated effects of accuracy (left) and precision (right) together with the line of significance for `data.old`. The definitions of the covariates A, B and C are given in (4.2).

The interpretation of Figure 4.31 is that decreasing the interest rate has a unfortunate impact on the model precision, while decreasing the amount of cash withdrawn from today's value to 0 contributes to a lower precision rate. Both of these observations are consistent with the results from the newer dataset.

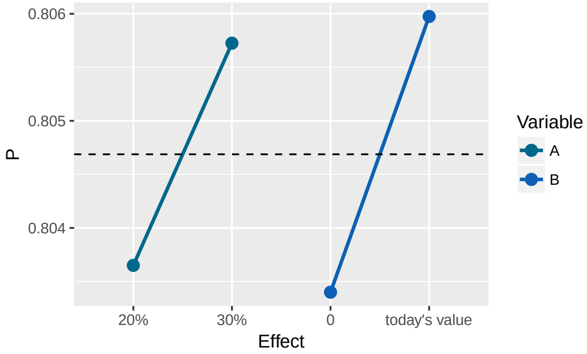


Figure 4.31: Significant main effects on precision for `data.old`

Chapter 5

Summary and Concluding Remarks

The aim of this thesis was to construct statistical models with high predictive power in detecting whether defaulted credit card customers recover or not. Several logistic regression models were fitted by including covariates using both backward elimination and regularization by elastic nets. An optimal support vector machine model was found by minimizing the residual sum of squares with respect to the included hyperparameters. The unsupervised approach of fuzzy clustering was also tested with the aim to investigate whether the data could be grouped such that each cluster shared a common recovery rate. This chapter gives an overview of the achieved goals and some recommendations of further studies.

5.1 Predictive Performance of the Supervised Models

Separating the dataset in two based on how long the accounts have been operative, we observed that the models fitted on the older accounts attained a higher accuracy and precision than the models fitted on the newer accounts. Analyzing the accuracy and precision of the logistic regression models and the support vector machine models revealed a decent predictive performance for both methods. Both the optimal accuracy and precision values were obtained with the regularized logistic model for the newer accounts, and the reduced logistic model for the older accounts. Their respective values were calculated to be $AC_{\text{new}} = 0.689, AC_{\text{old}} = 0.802$ and $P_{\text{new}} = 0.796, P_{\text{old}} = 0.815$. Neither the accuracy nor precision were proven to be statistically different between the reduced and regularized models for the newer accounts, while the precision proved to be statistically better in the reduced model for the older accounts.

The number of misclassified low-risk cases was unfortunately higher for the models of the older accounts compared to the newer accounts. The reason for this is believed to be the imbalance in the data as only 20.36% of the observations had a response value of RR $= 0$ in the older dataset, compared to 39.97% for the newer dataset. The distribution of predictions for the 0- and 1- cases were highly overlapping for several models. Exercising a soft threshold, with the aim of containing the overlapping observations, improved both the accuracy and precision for all models. The number of observations contained inside this threshold interval varied between 50% and 70%, stressing the amount of overlap.

5.2 Behavioural Trend in High-Risk Accounts

Investigating significant covariates in the logistic model of the newer accounts, revealed the age of the account, the number of payments and the number of times there has been an overdraft on the account as important. The longer an account has been operative increases the probability of recovering, which support the hypothesis of high-risk accounts defaulting more rapidly compared to low-risk accounts. A high number of overdrafts cause for a lower probability of recovering and transferring money to the credit card often will increase the probability of recovering.

When fitting the logistic model to the behaviour of the older accounts, the recorded average recovery rate for the past year and the number of months the account has had the status *normal* were considered important. An increased past RR caused an increased predicted value of the present RR, which was anticipated. The effect of the number of normal months was, on the other hand, interestingly estimated to be negative for both the older and newer accounts. This estimate could either be the result of laziness or the scare-off effect caused by the status *payment reminder*.

Comparing the support vectors with the remaining observations from the SVM for the newer accounts suggested that a typical high-risk behaviour is increasing the spending amount drastically from the first active month to the next. The mean value of the customer age was higher for the non-support vectors of the recovering accounts compared to the non-support vectors of the non-recovering accounts, implying higher financial stability for low-risk accounts. This observation was also made for the older accounts. For the older accounts, we also discovered that the number of previous defaults was highest for the support vectors which had RR $= 1$, meaning that accounts which often recover from default have a higher probability of recovering again.

The clustering algorithms did not obtain a satisfactory result. However, investigating the clusters that had a dominating fraction of recovering accounts revealed several dominating trends of the recovering accounts. Newer accounts which have an average incurred interest lower than 25 NOK, a maximal amount of cash withdrawals of less than 5% of the credit limit, and had not an evaluation method of type *ordinary* have a 91.6% chance of recovering. Hence, an increased amount of incurred interest and cash withdrawals causes a higher probability of not recovering. For the older accounts, a typical low-risk behaviour was revealed to be accounts which have a low balance sent to collection, infrequently receive payment reminders, rarely increase their credit limit, low amount of incurred interest and number of overdrafts. Among the accounts which satisfy MTPCollectionWarning < 350, SumDunning ≤ 4, SumCreditIncrease ≤ 1, AvgInterest < 150, AvgOverLimit < 0.2, 91.2% were recovering. Hence, the clusters were successful in detecting parts of the recovering cases but found it more difficult to distinguish the non-recovering cases.

5.3 Recommendations for Further Work

Only a discrete definition of RR was utilized in this thesis which causes loss of information. That is, customers who are able to restore a high fraction of their debt presents a lower risk compared to other recovering customers who are only able to pay the minimum amount required. The drawback is therefore that a discrete RR will not detect this difference of risk. By imposing the continuous definition of the RR, other methods such as beta regression could be considered [5, 45, 46]. Another approach would be to adopt a two-stage modelling framework similar to [4, 6] on the data presented in this thesis to better separate the 0- and 1-cases from the observations lying between the two extremes.

One possible reason to why the models are not successful in separating the low-risk cases from the high-risk cases is due to lack of information. The data provided for this thesis is solely based on transactional behaviour in addition to general information such as gender, age and income. This data does not contain enough information about why people act the way they do. We observe that a customer all of a sudden stops paying their bills without knowing they are going through a divorce or that they lost their job. Additionally, a customer in a comfortable financial situation could display a lack of commitment solely because paying their bills rarely is more appealing than saving money. It is not only external factors influencing people's choices but also human factors which are not available to the conventional bank. Today, separate companies possess different types of personal information, and there does not exist a common platform to incorporate all this information into one database. In the corporate world, striving for competitive advantage and obligations regarding the distribution of sensitive information, make this task even more difficult and maybe not ethically responsible. Nevertheless, combining information from e.g insurance-, health, social media- and bank companies could dramatically improve all analyses targeted to capture how users behave.

References

[1] J. MacDonald, T. Tompkins, (2017), *The history of credit cards*, `https://www.creditcards.com/credit-card-news/history-of-credit-cards.php`

[2] TSS, (2016), *2016 U.S Consumer Payment Study*, Total System Services, Inc.

[3] Basel Committee on Banking Supervision, (2005), *An explanatory note on the Basel II IRB risk weight functions*, Bank for International Settlements

[4] R.C Hwang, H. Chung, C. K. Chu, (2014), *A Two-Stage Probit Model for Predicting Recovery Rates*, Journal of Financial Services Research

[5] A. Moore, (2017), *Predicting Recovery Rates for Defaulting Credit Card Debt*, Quantitative Financial Risk Management Centre, University of Southamption

[6] X. Yao, J. Crook, G. Andreeva, (2017), *Enhancing two-stage modelling methodology for lossgiven default with support vector machines*, European Journal of Operational Research, Elsevier

[7] G. James, D. Witten, T. Hastie, R. Tibshirani, (2017), *An introduction to Statistical Learning with Applications in R*, Springer

[8] L. Fahrmeir, T. Kneib, S. Lang, B. Marx, (2013), *Regression Models, Methods and Applications*, Springer

[9] S. Le Cessie, J. C. van Houwelingen, (1992), *Ridge estimators in logistic regression*, Applied Statistics

[10] T. Hastie, R. Tibshirani, M. Wainwright, (2015), *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis

[11] I. Frank, J. Friedman, (1993), *A statistical view of some chemometrics regression tools*, Technometrics

[12] H. Zou, T. Hastie, (2005), *Regularization and variable selection via the elastic net*, Journal of Computational and Graphical Statistics

[13] J. Friedman, T. Hastie, R. Tibshirani, (2008) *Regularization Paths for Generalized Linear Models via Coordinate Descent*, Journal of Statistical Software

[14] P. D. Allison, (2014), *Measures of Fit for Logistic Regression*, University of Pennsylvania and Statistical Horizons LLC

[15] T. A. Domencich, D. McFadden, (1975), *Urban Travel Demand: A Behavioral Analysis*, North-Holland Publishing Company

[16] A. Agresti, (2002), *Categorical Data Analysis*, 2nd ed. John Wiley & Sons, Inc., Publication

[17] A. J. Dobson, (2002), *An Introduction to Generalized Linear Models*, 2nd ed. Chapman & Hall/CRC

[18] P. Paul, M. L. Pennell, S. Lemeshow, (2012), *Standardizing the power of the Hosmer Lemeshow goodness of fit test in large data sets*, The Ohio State University

[19] A. A. M. Nurunnabi, M. Nasser, (2011), *Outlier Diagnostics in Logistic Regression: A Supervised Learning Technique*, IACSIT Press, Singapore

[20] R. R. Bouckaert, (2003), *Choosing between Two Learning Algorithms based on Calibrated Tests*, Twentieth International Conference on Machine Learning

[21] T. Fawcett, (2005), *An Introduction to ROC Analysis*, Elsevier

[22] T. Fletcher, (2009), *Support Vector Machines Explained*, The UCL Department of Computer Science

[23] T. Hastie, R. Tibshirani, J. Friedman, (2016), *The elements of statistical learning*, Springer Series in Statistics

[24] G. Vining, S. M. Kowalski, 3rd ed. (2011), *Statistical Methods for Engineers*, Brooks/Cole, Cengage Learning

[25] NIST/SEMATECH, *e-Handbook of Statistical Methods*, `http://www.itl.nist.gov/div898/handbook/`, 2013.

[26] J. C. Bezdek, (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Springer

[27] M. Sato-Ilic, L. C. Jain, (2006), *Innovations in Fuzzy Clustering*, Springer

[28] E. E. Gustafson, W. C. Kessel, (1979), *Fuzzy clustering with a fuzzy covariance matrix*, Proc. IEEE CDC, San Diego, CA

[29] R. Krishnapuram, C. P. Freg, (1992), *Fitting an unknown number of lines and planes to image data through compatible cluster merging*, Pattern Recognition

[30] R. Xu, D. C. Wunsch, (2009), *Clustering*, John Wiley & Sons, Inc.

[31] D. Arthur, S. Vassilvitskii, (2007), *k-means++: The Advantages of Careful Seeding*, Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms

[32] M. B Al-Daoud, (2007), *A New Algorithm for Cluster Initialization*, Proc. of 2nd World Enformatika Conf.

[33] R. Krishnapuram, J. M. Keller, (1993), *A Possibilistic Approach to Clustering*, IEEE Transactions on fuzzy systems

[34] R. Winkler, F. Klawonn, R. Kruse, (2012), *Problems of Fuzzy c-Means Clustering and Similar Algorithms with High Dimensional Data Sets*, In: Gaul W., Geyer-Schulz A., Schmidt-Thieme L., Kunze J. (eds) Challenges at the Interface of Data Analysis, Computer Science, and Optimization. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Berlin, Heidelberg

[35] R. Winkler, F. Klawonn, R. Kruse (2011) *Fuzzy C-Means in High Dimensional Spaces*, International Journal of Fuzzy System Applications

[36] N. R. Pal, K. Pal, J. M. Keller, J. C. Bezdek, (2005), *A Possibilistic Fuzzy c-Means Clustering Algorithm*, IEEE Transactions on fuzzy systems

[37] I. Gath, A. B. Geva, (1989), *Unsupervised optimal fuzzy clustering*, IEEE Transaction on Pattern Analysis and Machine Intelligence

[38] J. C. Bezdek, (1974), *Cluster Validity with fuzzy sets*, J. Cybernetics

[39] J. C. Bezdek, J. C. Dunn, (1975), *Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions*, IEEE Transaction on Computers

[40] X. L. Xie, G. Beni, (1991), *A validity measure for fuzzy clustering*, IEEE Trans. Pattern Anal. Mach. Intell.

[41] S. Yang, K. Li, Z. Liang, W. Li, Y. Xue, (2016), *A new method of choosing the number of clusters for the fuzzy c-means method*, Springer

[42] T. Bellotti, J. Crook, (2012), *Loss given default models incorporating macroeconomic variables for credit cards*, International Journal of Forecasting

[43] A. Karatzoglou, D. Meyer, K. Hornik, (2006), *Support Vector Machines in R*, Journal of Statistical Software

[44] W. Martin, (2018), *Fewer than 10% of people in Norway use cash — and a senior official thinks it could disappear completely in a decade*, Business Insider Nordic

[45] A. Morozovskiy, (2007), *Why Beta-Distribution - Demand/Supply Theory of Recovery Rates*, Available at SSRN: `https://ssrn.com/abstract=958150`

[46] R. Chen, Z. Wang, (2012), *The Comparison of Beta Distribution Estimation and Gauss Kernel Density Estimation in the Recovery Rates of Municipal Bonds*, Fifth International Conference on Business Intelligence and Financial Engineering, Lanzhou

Appendix A

Data

Table A.1: Variables included in the data set provided by Sparebank 1

Variable	Explanation
BK_COLLECTION_CASE_ID	Key, unique for each collection case
Recovery	If account recovered = 1, else = 0
DebtCollectionCompany	Which debt collection company (GOT, CON, LIN) handled the case
RecruitmentChannel	Where the account has been sold, ex. online, mobile bank, operation channel etc.
AgeOfCustomer	Age of customer in years
Gender	Gender of customer
AgeOfACcount	Age of account in months
Bank	Which bank does the account belong to
SerialN	Digits from social security number
Product	Type of credit card product
ClosingBalanceCollectionWarning	Closing balance at invoice for collection warning
MTPCollectionWarning	Minimum payment required for recovery, and is calculated from ClosingBalanceCollectionWarning
SumCollectionWarning	Number of times the account has been sent a collection warning
SumDunning	Number of times the account has been sent a dunning
Recurring	Number of times the account has been sent to collection warning previously
EvaluationMethod	How the account has been sold, ex. through mortgage, campaign, ordinary, platinum, young/student etc.
DebtAppl	Current debt given in the credit card application
GrossIncomeAppl	Gross income for current year given in credit card application
GrossIncome	Actual registered gross income for current year
ClosingBalance	Closing balance for each month
ShareClosingBalance	Closing balance divided by credit card limit for each month
PayedShare	Payments divided with closing balance
CredLimit	Credit limit each month
CredLimitIncrease	Amount the credit limit has increased with from the previous month
OverLimit	If the balance during the month has exceeded the credit limit = 1, else = 0
PayedAmt	Amount payed each month
PayedN	Number of payments each month
Interest	Incurred interest each month
Purchase	Amount spent on purchases each month
CashWithdrawal	Amount of cash withdrawn each month
CashTransfer	Amount transferred to another bank account each month
InvoiceType	Type of invoice, ex. normal, payment reminder, collection warning etc.

Table A.2: Variables added to the data set

Variable	**Explanation**
`Active`	Number of moths the credit card has been used
`RR`	Total recovery rate for the previously recorded months, i.e the sum of payments divided by the sum of closing balance for the last six months
`RR.avg`	Average recovery rate for the previously recorded months, i.e sum of payments divided by closing balance each month
`AvgClosingBalance`	Average closing balance for the active months
`SumCollectionWarning`	Total number of collection warnings for the last six months
`SumDunning`	Total number of dunnings for the last six months
`SumCreditIncrease`	Total number of times the credit limit has been increased during the last six months
`StartGrad`	Closing balance divided by the credit limit for the first active month
`AvgNormal`	Average number of times the invoice has had status ”normal” for the active months
`AvgCashTransfer`	Average cash transfers for the active months
`AvgPurchase`	Average purchases for the active months
`AvgInterest`	Average incurred interest for the active months
`AvgPayN`	Average number of payments for the active months
`AvgOverLimit`	Average number of times the credit balance has exceeded the credit limit for the active months
`MaxClosingBalance`	Maximum closing balance
`DaysSinceLastTime`	Days since the last time the account was sent to collection (if the account has never been sent to collection before, then its value is set to 9999)
`AvgCashWithdrawal`	Average cash withdrawals for the active months
`AvgLessThanMin`	Average number of times the payment was less than 5% of the closing balance
`FirstBalance`	First active balance
`FirstActive`	Number of months between receiving the credit card and usage
`Returned`	If the customer has returned from not paying or has always payed $= 1$, if the customer has never payed $= 0$, if the customer has not returned -1
`SumOverLimit`	Total number of times the credit balance has exceeded the credit limit
`MaxCashTransfer`	Maxmimum cash transfer
`MaxCashWithdrawal`	Maximum cash withdrawal

Appendix B

Results

Table B.1: Accuracy, precision and AUC for the final logistic - and SVM models presented in the same order as in section 4.

Model	Data set	Threshold	AC	P	AUC
Reduced Logistic	`data.new`	0.5 $[0.2, 0.8]$	0.679 0.866	0.710 0.864	0.751
Regularized Logistic	`data.new`	0.5 $[0.3, 0.75]$	0.689 0.898	0.796 0.898	0.746
Reduced Logistic	`data.old`	0.5 $[0.3, 0.85]$	0.802 0.905	0.815 0.909	0.737
Regularized Logistic	`data.old`	0.5 $[0.3, 0.85]$	0.799 0.908	0.809 0.911	0.733
SVM	`data.new`	0.5 $[0.2, 0.8]$	0.687 0.863	0.704 0.858	0.734
	`data.old`	0.5 $[0.75, 0.85]$	0.799 0.855	0.807 0.912	0.718

Full Logistic Model for `data.new`

```
Call:
glm(formula = Recovery ~ ., family = "binomial", data = english.yng[train_ind,
    -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7878  -0.9517   0.4191   0.8770   2.7059

Coefficients: (2 not defined because of singularities)
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                            3.554e+00  1.076e+03   0.003 0.997364
DebtCollectionCompanyGOT               2.117e-01  9.907e-02   2.137 0.032617 *
```

```
DebtCollectionCompanyLIN              1.232e-01  1.860e-01   0.662 0.507660
ProductLOfavoer MasterCard             -1.225e+00  9.349e-01  -1.310 0.190100
ProductSB1 EXTRA MC                  -2.035e-01  6.490e-01  -0.314 0.753888
ProductSH GOLD MC                     1.817e-01  3.099e-01   0.586 0.557603
ProductSpareBank 1 MasterCard Gold    2.902e-01  5.690e-01   0.510 0.610103
ProductSparebank 1 Platinum MC        2.301e+00  9.475e+02   0.002 0.998063
ProductSpareBank 1 Visa Business Card -9.386e+00  8.827e+02  -0.011 0.991517
ProductSpareBank 1 Visa Gold          2.622e-01  9.658e-01   0.271 0.786018
BankSpareBank 1 BV                    1.762e-01  5.549e-01   0.318 0.750795
BankSpareBank 1 Gudbrandsdal          1.057e+00  7.722e-01   1.368 0.171160
BankSpareBank 1 Hallingdal Valdres    4.705e-01  7.219e-01   0.652 0.514511
BankSpareBank 1 Lom og Skjaak        -1.141e-01  7.968e-01  -0.143 0.886090
BankSpareBank 1 Modum                -2.826e-02  6.126e-01  -0.046 0.963199
BankSpareBank 1 Nord-Norge            2.432e-01  5.251e-01   0.463 0.643285
BankSpareBank 1 Nordvest             -2.697e-01  5.959e-01  -0.453 0.650794
BankSpareBank 1 Noetteroey-Toensberg  5.946e-01  7.020e-01   0.847 0.396971
BankSpareBank 1 Oslo Akershus        -1.153e-02  5.248e-01  -0.022 0.982465
BankSpareBank 1 Oestfold Akershus     1.182e-01  5.445e-01   0.217 0.828111
BankSpareBank 1 oestlandet           -4.131e-01  5.816e-01  -0.710 0.477474
BankSpareBank 1 Ringerike Hadeland    6.141e-02  5.832e-01   0.105 0.916148
BankSpareBank 1 SMN                   1.335e-01  5.247e-01   0.254 0.799134
BankSpareBank 1 Soere Sunnmoere       6.827e-02  6.667e-01   0.102 0.918442
BankSpareBank 1 SR-Bank              -6.970e-02  5.202e-01  -0.134 0.893419
BankSpareBank 1 Telemark             -3.830e-01  5.662e-01  -0.676 0.498753
BankSparebanken Hedmark                      NA         NA      NA       NA
SerialN10                             1.291e+01  6.146e+02   0.021 0.983246
SerialN20                             1.342e+01  6.146e+02   0.022 0.982583
SerialN30                             1.400e+01  6.146e+02   0.023 0.981831
SerialN40                             1.398e+01  6.146e+02   0.023 0.981854
AgeOfAccount                          4.511e-01  5.465e-01   0.825 0.409131
GenderM                              -1.628e-01  9.077e-02  -1.793 0.072940 .
AgeOfCustomer                        -4.074e-03  4.383e-03  -0.930 0.352622
RecruitmentChannelLO Channel          6.893e-01  3.209e-01   2.148 0.031711 *
RecruitmentChannelMobilebank          5.265e-02  3.445e-01   0.153 0.878509
RecruitmentChannelOnlinebank          1.681e-01  1.603e-01   1.049 0.294326
RecruitmentChannelNULL                1.314e+01  6.184e+02   0.021 0.983044
RecruitmentChannelOpen web           -1.147e-01  1.022e+00  -0.112 0.910670
RecruitmentChannelOperationchannel    6.319e-01  1.741e-01   3.630 0.000284 ***
RecruitmentChannelResponsepage        3.537e-01  2.052e-01   1.724 0.084782 .
MTPCollectionWarning                 -1.898e-04  5.389e-05  -3.522 0.000428 ***
ClosingBalanceCollectionWarning       5.084e-05  3.397e-05   1.497 0.134481
SumCollectionWarning                 -7.000e-02  3.785e-01  -0.185 0.853265
SumDunning                           -1.968e-01  3.373e-01  -0.583 0.559612
Recurring                            -7.659e-01  1.271e+00  -0.602 0.546921
SumCreditIncrease                     7.667e-01  4.122e-01   1.860 0.062871 .
DebtAppl                              1.239e-11  7.946e-11   0.156 0.876043
GrossIncomeAppl                      -1.037e-09  2.034e-09  -0.510 0.610208
```

```
GrossIncome                          9.913e-07  2.331e-07   4.253 2.11e-05 ***
EvaluationMethodMortgage            -1.263e+01  8.827e+02  -0.014 0.988580
EvaluationMethodCampaign            -1.253e+01  8.827e+02  -0.014 0.988678
EvaluationMethodNULL                -1.129e+01  8.827e+02  -0.013 0.989797
EvaluationMethodOrdinary            -1.296e+01  8.827e+02  -0.015 0.988287
EvaluationMethodPlatinum                    NA         NA      NA       NA
EvaluationMethodYoung/Student       -1.249e+01  8.827e+02  -0.014 0.988709
Active                              -5.486e-01  5.675e-01  -0.967 0.333704
RR                                   8.353e-01  1.426e+00   0.586 0.557929
RR.avg                               3.328e+00  2.863e+00   1.163 0.245031
AvgClosingBalance                   -8.995e-01  8.407e-01  -1.070 0.284676
StartGrad                            4.152e-01  3.826e-01   1.085 0.277915
AvgNormal                           -1.698e+00  1.434e+00  -1.184 0.236492
AvgCashTransfer                     -2.635e+00  1.677e+00  -1.571 0.116072
AvgPurchase                         -1.264e+00  1.535e+00  -0.823 0.410336
AvgInterest                         -1.510e-03  1.310e-03  -1.153 0.248877
AvgPayN                              1.517e+00  3.675e-01   4.127 3.67e-05 ***
AvgOverLimit                        -8.987e-01  4.658e-01  -1.929 0.053696 .
MaxClosingBalance                   -1.375e-05  3.099e-05  -0.444 0.657336
DaysSinceLastTime                   -1.015e-04  1.339e-04  -0.758 0.448434
CreditLimit                         -1.180e-05  1.467e-05  -0.805 0.421101
AvgCashWithdrawal                    1.924e+00  1.756e+00   1.096 0.273005
AvgLessThanMin                      -4.616e-01  5.356e-01  -0.862 0.388732
FirstBalance                        -1.725e-05  1.090e-05  -1.582 0.113698
FirstActive                         -2.402e-01  5.513e-01  -0.436 0.663037
Returned                             2.032e-01  8.557e-02   2.375 0.017566 *
SumOverLimit                        -1.766e-02  1.202e-01  -0.147 0.883218
MaxCashTransfer                      1.424e-01  3.192e-01   0.446 0.655499
MaxCashWithdrawal                   -6.039e-01  5.386e-01  -1.121 0.262194
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3752.2  on 2794  degrees of freedom
Residual deviance: 3000.7  on 2719  degrees of freedom
AIC: 3152.7

Number of Fisher Scoring iterations: 13
```

Reduced Logistic Model for `data.new`

```
Call:
glm(formula = Recovery ~ DebtCollectionCompany + SerialN + AgeOfAccount +
    Gender + RecruitmentChannel + MTPCollectionWarning + ClosingBalanceCollectionWarning
    SumCreditIncrease + GrossIncome + EvaluationMethod + Active +
    RR.avg + AvgNormal + AvgCashTransfer + AvgInterest + AvgPayN +
    AvgOverLimit + Returned, family = "binomial", data = english.yng[train_ind,
    -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7252  -0.9520   0.4346   0.8878   2.6456

Coefficients:
                                     Estimate Std. Error z value Pr(>|z|)
(Intercept)                        -9.256e+00  6.140e+02  -0.015 0.987973
DebtCollectionCompanyGOT            2.033e-01  9.771e-02   2.080 0.037505 *
DebtCollectionCompanyLIN            8.641e-02  1.816e-01   0.476 0.634225
SerialN10                           1.301e+01  6.140e+02   0.021 0.983095
SerialN20                           1.351e+01  6.140e+02   0.022 0.982446
SerialN30                           1.409e+01  6.140e+02   0.023 0.981691
SerialN40                           1.408e+01  6.140e+02   0.023 0.981707
AgeOfAccount                        2.005e-01  7.326e-02   2.736 0.006213 **
GenderM                            -1.348e-01  8.923e-02  -1.511 0.130740
RecruitmentChannelLO Channel        6.324e-01  3.067e-01   2.062 0.039190 *
RecruitmentChannelMobilebank        2.586e-01  3.301e-01   0.784 0.433305
RecruitmentChannelONlinebank        3.294e-01  1.410e-01   2.336 0.019486 *
RecruitmentChannelNULL              1.473e+01  6.164e+02   0.024 0.980938
RecruitmentChannelOpen web          8.132e-02  1.020e+00   0.080 0.936429
RecruitmentChannelOperationChannel  7.702e-01  1.609e-01   4.787 1.69e-06 ***
RecruitmentChannelResponsepage      5.078e-01  1.914e-01   2.653 0.007975 **
MTPCollectionWarning               -1.892e-04  4.997e-05  -3.787 0.000153 ***
ClosingBalanceCollectionWarning     2.619e-05  8.687e-06   3.015 0.002567 **
SumCreditIncrease                   7.874e-01  3.883e-01   2.028 0.042609 *
GrossIncome                         9.288e-07  2.275e-07   4.082 4.46e-05 ***
EvaluationMethodMortgage           -2.882e+00  1.619e+00  -1.780 0.075065 .
EvaluationMethodCampaign           -2.750e+00  1.603e+00  -1.716 0.086189 .
EvaluationMethodNULL               -2.927e+00  1.617e+00  -1.810 0.070220 .
EvaluationMethodOrdinary           -3.213e+00  1.604e+00  -2.003 0.045157 *
EvaluationMethodPlatinum            1.170e+01  3.516e+02   0.033 0.973454
EvaluationMethodYoung/Student      -2.759e+00  1.623e+00  -1.701 0.089001 .
Active                             -2.468e-01  9.237e-02  -2.672 0.007539 **
RR.avg                              2.771e+00  1.884e+00   1.471 0.141423
AvgNormal                          -1.143e+00  7.245e-01  -1.577 0.114820
AvgCashTransfer                    -1.276e+00  4.435e-01  -2.878 0.004007 **
AvgInterest                        -3.190e-03  8.234e-04  -3.875 0.000107 ***
AvgPayN                             1.547e+00  3.404e-01   4.544 5.53e-06 ***
AvgOverLimit                       -1.096e+00  1.420e-01  -7.715 1.21e-14 ***
Returned                            1.663e-01  7.921e-02   2.099 0.035807 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3752.2  on 2794  degrees of freedom
Residual deviance: 3037.6  on 2761  degrees of freedom
AIC: 3105.6

Number of Fisher Scoring iterations: 13
```

ANOVA for Full- and Reduced Model for `data.new`

```
Analysis of Deviance Table

Model 1: Recovery ~ DebtCollectionCompany + SerialN + AgeOfAccount + Gender +
    RecruitmentChannel + MTPCollectionWarning + ClosingBalanceCollectionWarning +
    SumCreditIncrease + GrossIncome + EvaluationMethod + Active +
    RR.avg + AvgNormal + AvgCashTransfer + AvgInterest + AvgPayN +
    AvgOverLimit + Returned
Model 2: Recovery ~ DebtCollectionCompany + Product + Bank + SerialN +
    AgeOfAccount + Gender + AgeOfCustomer + RecruitmentChannel +
    MTPCollectionWarning + ClosingBalanceCollectionWarning +
    SumCollectionWarning + SumDunning + Recurring + SumCreditIncrease +
    DebtAppl + GrossIncomeAppl + GrossIncome + EvaluationMethod +
    Active + RR + RR.avg + AvgClosingBalance + StartGrad + AvgNormal +
    AvgCashTransfer + AvgPurchase + AvgInterest + AvgPayN + AvgOverLimit +
    MaxClosingBalance + DaysSinceLastTime + CreditLimit + AvgCashWithdrawal +
    AvgLessThanMin + FirstBalance + FirstActive + Returned +
    SumOverLimit + MaxCashTransfer + MaxCashWithdrawal
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      2761     3037.6
2      2719     3000.7 42   36.934   0.6926
```

Regularized Model for `data.new`

```
Call:
glm(formula = Recovery ~ ., family = "binomial", data = reg.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6928  -0.9745   0.4500   0.9049   1.9893

Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                             2.541e+00  4.831e-01   5.259 1.45e-07 ***
DebtCollectionCompanyGOT                2.796e-01  9.421e-02   2.968 0.003000 **
```

```
ProductSB1.EXTRA.MC                    -7.632e-01  3.154e-01  -2.419 0.015549 *
SerialN10                              -7.143e-01  2.043e-01  -3.496 0.000473 ***
SerialN20                              -4.026e-01  9.828e-02  -4.096 4.20e-05 ***
RecruitmentChannelOperatationchannel    4.378e-01  1.043e-01   4.197 2.71e-05 ***
MTPCollectionWarning                   -1.016e-04  2.769e-05  -3.671 0.000242 ***
GrossIncome                             1.106e-06  2.201e-07   5.022 5.10e-07 ***
EvaluationMethodOrdinary               -5.423e-01  9.019e-02  -6.012 1.83e-09 ***
AvgClosingBalance                      -8.260e-01  3.761e-01  -2.196 0.028064 *
AvgInterest                            -9.033e-04  3.657e-04  -2.470 0.013509 *
AvgPayN                                 1.109e+00  2.593e-01   4.277 1.89e-05 ***
AvgOverLimit                           -1.055e+00  1.606e-01  -6.567 5.13e-11 ***
AvgLessThanMin                         -6.606e-01  4.296e-01  -1.537 0.124171
FirstActive                             2.455e-01  6.959e-02   3.528 0.000418 ***
Returned                                2.507e-01  6.494e-02   3.861 0.000113 ***
MaxCashWithdrawal                      -3.922e-01  1.565e-01  -2.507 0.012190 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3758.0  on 2794  degrees of freedom
Residual deviance: 3093.5  on 2778  degrees of freedom
AIC: 3127.5

Number of Fisher Scoring iterations: 5
```

ANOVA for Reduced- and Regularized Model for `data.new`

```
Analysis of Deviance Table

Model 1: Recovery ~ DebtCollectionCompanyGOT + ProductSB1.EXTRA.MC + SerialN10 +
    SerialN20 + RecruitmentChannelOperationchannel + MTPCollectionWarning +
    GrossIncome + EvaluationMethodOrdinary + AvgClosingBalance +
    AvgInterest + AvgPayN + AvgOverLimit + AvgLessThanMin + FirstActive +
    Returned + MaxCashWithdrawal
Model 2: Recovery ~ DebtCollectionCompany + SerialN + AgeOfAccount + Gender +
    RecruitmentChannel + MTPCollectionWarning + ClosingBalanceCollectionWarning +
    SumCreditIncrease + GrossIncome + EvaluationMethod + Active +
    RR.avg + AvgNormal + AvgCashTransfer + AvgInterest + AvgPayN +
    AvgOverLimit + Returned
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2778     3093.5
2      2761     3037.6 17   55.873 4.958e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hosmer-Lemeshow for `data.new`

```
        Hosmer and Lemeshow goodness of fit (GOF) test

data:  reduced.model$y, fitted(reduced.model)
X-squared = 65.136, df = 63, p-value = 0.4023
```

Full Logistic Model for `data.old`

```
Call:
glm(formula = Recovery ˜ ., family = "binomial", data = english.old[train_ind,
    -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9302   0.2958   0.4882   0.6778   3.5367

Coefficients: (1 not defined because of singularities)
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                            3.329e+00  1.109e+00   3.001 0.002691 **
DebtCollectionCompanyGOT               1.538e-01  3.573e-02   4.304 1.67e-05 ***
DebtCollectionCompanyLIN              -1.327e-01  5.933e-02  -2.236 0.025323 *
ProductLOfavoer MasterCard            -5.343e-01  3.382e-01  -1.580 0.114114
ProductSB1 EXTRA MC                   -1.079e+00  3.130e-01  -3.448 0.000566 ***
ProductSH BUSINESS VISA                1.291e+00  1.041e+00   1.240 0.215052
ProductSH GOLD MC                     -9.781e-02  1.276e-01  -0.766 0.443443
ProductSpareBank 1 MasterCard Gold    -1.845e-01  2.385e-01  -0.774 0.439117
ProductSparebank 1 Platinum MC         3.315e-01  4.045e-01   0.820 0.412479
ProductSpareBank 1 Visa Business Card  1.743e+00  6.454e-01   2.700 0.006926 **
ProductSpareBank 1 Visa Gold          -5.654e-01  2.488e-01  -2.273 0.023049 *
BankSpareBank 1 BV                     2.984e-01  2.263e-01   1.319 0.187310
BankSpareBank 1 Gudbrandsdal           9.269e-01  2.988e-01   3.102 0.001922 **
BankSpareBank 1 Hallingdal Valdres     4.643e-01  2.669e-01   1.740 0.081935 .
BankSpareBank 1 Kredittkort           -2.761e+00  7.171e-01  -3.850 0.000118 ***
BankSpareBank 1 Lom og Skjaak          7.930e-01  3.306e-01   2.399 0.016452 *
BankSpareBank 1 Modum                  3.371e-01  2.580e-01   1.307 0.191382
BankSpareBank 1 Nord-Norge             5.147e-01  2.176e-01   2.366 0.017981 *
BankSpareBank 1 Nordvest               7.573e-01  2.435e-01   3.110 0.001868 **
BankSpareBank 1 Noetteroey-Toensberg   4.781e-01  2.647e-01   1.806 0.070932 .
BankSpareBank 1 Oslo Akershus          2.768e-01  2.181e-01   1.269 0.204407
BankSpareBank 1 oestfold Akershus      4.338e-01  2.259e-01   1.920 0.054857 .
BankSpareBank 1 oestlandet             1.614e-01  1.672e-01   0.965 0.334591
BankSpareBank 1 Ringerike Hadeland     3.220e-01  2.349e-01   1.371 0.170417
BankSpareBank 1 SMN                    4.449e-01  2.170e-01   2.051 0.040306 *
BankSpareBank 1 Soere Sunnmoere        7.691e-01  2.669e-01   2.882 0.003953 **
BankSpareBank 1 SR-Bank              3.432e-01  2.161e-01   1.588 0.112178
BankSpareBank 1 Telemark             4.350e-01  2.318e-01   1.876 0.060621 .
BankSparebanken Hedmark                     NA         NA      NA       NA
SerialN10                           -2.209e-01  6.286e-01  -0.351 0.725303
SerialN20                            4.870e-02  6.246e-01   0.078 0.937859
SerialN30                            3.825e-01  6.243e-01   0.613 0.540065
SerialN40                            4.949e-01  6.243e-01   0.793 0.427987
AgeOfAccount                         3.555e-03  7.439e-04   4.779 1.76e-06 ***
GenderM                             -3.677e-02  3.223e-02  -1.141 0.253989
AgeOfCustomer                       -8.508e-03  1.570e-03  -5.421 5.93e-08 ***
RecruitmentChannelLO Channel         1.032e+00  3.097e-01   3.333 0.000858 ***
RecruitmentChannelMobilebank         7.758e-01  4.824e-01   1.608 0.107764
RecruitmentChannelOnlinebank         3.740e-01  9.900e-02   3.777 0.000159 ***
RecruitmentChannelNULL               7.613e-01  1.819e-01   4.186 2.85e-05 ***
RecruitmentChannelOpen web           7.174e-02  5.332e-01   0.135 0.892973
RecruitmentChannelOperationChannel   6.960e-01  9.917e-02   7.018 2.24e-12 ***
RecruitmentChannelResponsepage       4.370e-01  1.148e-01   3.807 0.000141 ***
MTPCollectionWarning                -1.307e-04  1.243e-05 -10.518  < 2e-16 ***
ClosingBalanceCollectionWarning      1.351e-05  3.232e-06   4.181 2.90e-05 ***
SumCollectionWarning                 3.993e-02  2.150e-02   1.857 0.063301 .
SumDunning                           1.016e-01  9.625e-03  10.550  < 2e-16 ***
Recurring                           -5.438e-02  2.661e-02  -2.044 0.040974 *
SumCreditIncrease                   -1.354e-01  2.762e-02  -4.905 9.36e-07 ***
PaymentMethod NULL                  -2.897e+00  8.719e-01  -3.323 0.000891 ***
PaymentMethod Print                  3.403e-01  3.913e-02   8.698  < 2e-16 ***
DebtAppl                             1.471e-07  3.565e-08   4.127 3.68e-05 ***
GrossIncomeAppl                     -2.780e-09  6.077e-09  -0.458 0.647289
GrossIncome                          6.938e-07  1.099e-07   6.314 2.72e-10 ***
EvaluationMethodMortgage            -8.791e-01  8.768e-01  -1.003 0.316067
EvaluationMethodCampaign            -1.315e+00  8.711e-01  -1.510 0.131006
EvaluationMethodNULL                -1.350e+00  8.816e-01  -1.531 0.125704
EvaluationMethodOrdinary            -1.568e+00  8.705e-01  -1.801 0.071742 .
EvaluationMethodPlatinum            -1.535e+00  1.082e+00  -1.419 0.155951
EvaluationMethodYOung/Student       -1.495e+00  8.795e-01  -1.699 0.089241 .
Active                              -4.429e-02  1.704e-02  -2.598 0.009364 **
RR                                   1.293e-01  1.135e-01   1.139 0.254528
RR.avg                               1.146e+00  2.929e-01   3.911 9.18e-05 ***
AvgClosingBalance                   -4.313e-01  1.229e-01  -3.511 0.000446 ***
AvgNormal                           -1.594e+00  1.812e-01  -8.792  < 2e-16 ***
AvgCashTransfer                      7.301e-03  3.854e-01   0.019 0.984886
AvgPurchase                          2.219e-01  3.185e-01   0.697 0.485981
AvgInterest                         -5.712e-04  1.224e-04  -4.665 3.09e-06 ***
AvgPayN                              7.014e-01  1.231e-01   5.697 1.22e-08 ***
AvgOverLimit                        -7.682e-01  6.893e-02 -11.145  < 2e-16 ***
DaysSinceLastTime                    1.454e-05  4.917e-06   2.958 0.003095 **
CreditLimit                         -5.322e-06  2.387e-06  -2.229 0.025786 *
AvgCashWithdrawal                   -1.295e+00  7.191e-01  -1.801 0.071711 .
```

```
AvgLessThanMin                        -2.573e-01  6.466e-02  -3.979 6.92e-05 ***
MaxCashTransfer                       -3.980e-01  7.942e-02  -5.012 5.39e-07 ***
MaxCashWithdrawal                     -2.458e-01  1.350e-01  -1.820 0.068730 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27809  on 27580  degrees of freedom
Residual deviance: 24399  on 27505  degrees of freedom
AIC: 24551

Number of Fisher Scoring iterations: 10
```

Reduced Logistic Model for `data.old`

```
Call:
glm(formula = Recovery ˜ DebtCollectionCompany + Product + Bank +
    SerialN + AgeOfAccount + AgeOfCustomer + RecruitmentChannel +
    MTPCollectionWarning + ClosingBalanceCollectionWarning +
    SumCollectionWarning + SumDunning + Recurring + SumCreditIncrease +
    ‘PaymentMethod ‘ + DebtAppl + GrossIncome + EvaluationMethod +
    Active + RR.avg + AvgClosingBalance + AvgNormal + AvgInterest +
    AvgPayN + AvgOverLimit + DaysSinceLastTime + CreditLimit +
    AvgCashWithdrawal + AvgLessThanMin + MaxCashTransfer + MaxCashWithdrawal,
    family = "binomial", data = english.old[train_ind, -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9305   0.2963   0.4887   0.6765   3.5413

Coefficients: (1 not defined because of singularities)
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                            3.389e+00  1.108e+00   3.059 0.002223 **
DebtCollectionCompanyGOT               1.528e-01  3.571e-02   4.278 1.89e-05 ***
DebtCollectionCompanyLIN              -1.333e-01  5.931e-02  -2.247 0.024659 *
ProductLOfavoer MasterCard            -5.313e-01  3.378e-01  -1.573 0.115772
ProductSB1 EXTRA MC                   -1.080e+00  3.130e-01  -3.450 0.000560 ***
ProductSH BUSINESS VISA                1.268e+00  1.039e+00   1.221 0.222248
ProductSH GOLD MC                     -1.002e-01  1.276e-01  -0.785 0.432483
ProductSpareBank 1 MasterCard Gold    -1.839e-01  2.385e-01  -0.771 0.440692
ProductSparebank 1 Platinum MC         3.267e-01  4.038e-01   0.809 0.418471
ProductSpareBank 1 Visa Business Card  1.712e+00  6.424e-01   2.666 0.007681 **
ProductSpareBank 1 Visa Gold          -5.646e-01  2.488e-01  -2.269 0.023240 *
BankSpareBank 1 BV                     2.972e-01  2.264e-01   1.313 0.189217
BankSpareBank 1 Gudbrandsdal           9.248e-01  2.988e-01   3.095 0.001967 **
BankSpareBank 1 Hallingdal Valdres     4.610e-01  2.669e-01   1.727 0.084159 .
BankSpareBank 1 Kredittkort           -2.768e+00  7.170e-01  -3.861 0.000113 ***
BankSpareBank 1 Lom og Skjaak          7.875e-01  3.305e-01   2.383 0.017192 *
BankSpareBank 1 Modum                  3.342e-01  2.580e-01   1.295 0.195242
BankSpareBank 1 Nord-Norge             5.129e-01  2.176e-01   2.357 0.018412 *
BankSpareBank 1 Nordvest               7.552e-01  2.435e-01   3.102 0.001923 **
BankSpareBank 1 Noetteroey-Toensberg   4.788e-01  2.647e-01   1.809 0.070441 .
BankSpareBank 1 Oslo Akershus          2.752e-01  2.181e-01   1.261 0.207173
BankSpareBank 1 oestfold Akershus      4.287e-01  2.259e-01   1.898 0.057752 .
BankSpareBank 1 oestlandet             1.594e-01  1.673e-01   0.953 0.340640
BankSpareBank 1 Ringerike Hadeland     3.188e-01  2.349e-01   1.357 0.174787
BankSpareBank 1 SMN                    4.421e-01  2.170e-01   2.038 0.041593 *
BankSpareBank 1 Soere Sunnmoere        7.667e-01  2.669e-01   2.873 0.004069 **
BankSpareBank 1 SR-Bank                3.405e-01  2.161e-01   1.576 0.115054
BankSpareBank 1 Telemark               4.309e-01  2.318e-01   1.859 0.063087 .
BankSparebanken Hedmark                       NA         NA      NA       NA
SerialN10                             -2.266e-01  6.286e-01  -0.361 0.718425
SerialN20                              4.433e-02  6.246e-01   0.071 0.943413
SerialN30                              3.803e-01  6.243e-01   0.609 0.542352
SerialN40                              4.938e-01  6.243e-01   0.791 0.428942
AgeOfAccount                           3.521e-03  7.434e-04   4.736 2.18e-06 ***
AgeOfCustomer                         -8.539e-03  1.566e-03  -5.453 4.95e-08 ***
RecruitmentChannelLO Channel           1.025e+00  3.094e-01   3.313 0.000923 ***
RecruitmentChannelMobilebank           7.830e-01  4.820e-01   1.624 0.104281
RecruitmentChannelOnlinebank           3.733e-01  9.901e-02   3.770 0.000163 ***
RecruitmentChannelNULL                 7.630e-01  1.819e-01   4.196 2.72e-05 ***
RecruitmentChannelOpen web             6.094e-02  5.333e-01   0.114 0.909026
RecruitmentChannelOperationchannel     6.966e-01  9.918e-02   7.024 2.16e-12 ***
RecruitmentChannelResponsepage         4.352e-01  1.148e-01   3.791 0.000150 ***
MTPCollectionWarning                  -1.302e-04  1.238e-05 -10.519  < 2e-16 ***
ClosingBalanceCollectionWarning        1.423e-05  2.294e-06   6.204 5.51e-10 ***
SumCollectionWarning                   4.072e-02  2.149e-02   1.895 0.058079 .
SumDunning                             1.017e-01  9.611e-03  10.578  < 2e-16 ***
Recurring                             -5.572e-02  2.657e-02  -2.097 0.035989 *
SumCreditIncrease                     -1.338e-01  2.712e-02  -4.934 8.05e-07 ***
PaymentMethod NULL                    -2.917e+00  8.727e-01  -3.342 0.000831 ***
PaymentMethod Print                    3.406e-01  3.909e-02   8.713  < 2e-16 ***
DebtAppl                               1.464e-07  3.551e-08   4.122 3.76e-05 ***
GrossIncome                            6.943e-07  1.098e-07   6.324 2.55e-10 ***
EvaluationMethodMortgage              -9.020e-01  8.776e-01  -1.028 0.304045
EvaluationMethodCampaign              -1.338e+00  8.719e-01  -1.535 0.124763
EvaluationMethodNULL                  -1.376e+00  8.824e-01  -1.559 0.118922
EvaluationMethodOrdinary              -1.591e+00  8.713e-01  -1.826 0.067838 .
EvaluationMethodPlatinum              -1.543e+00  1.083e+00  -1.425 0.154093
EvaluationMethodYoung/Student         -1.512e+00  8.803e-01  -1.717 0.085920 .
Active                                -4.843e-02  1.657e-02  -2.923 0.003472 **
RR.avg                                 1.366e+00  2.441e-01   5.595 2.21e-08 ***
```

```
AvgClosingBalance                     -4.375e-01  1.213e-01  -3.606 0.000311 ***
AvgNormal                             -1.584e+00  1.804e-01  -8.782  < 2e-16 ***
AvgInterest                           -5.610e-04  1.180e-04  -4.755 1.99e-06 ***
AvgPayN                                7.138e-01  1.225e-01   5.825 5.70e-09 ***
AvgOverLimit                          -7.691e-01  6.842e-02 -11.240  < 2e-16 ***
DaysSinceLastTime                      1.469e-05  4.912e-06   2.990 0.002788 **
CreditLimit                           -5.069e-06  2.073e-06  -2.445 0.014479 *
AvgCashWithdrawal                     -1.224e+00  6.438e-01  -1.901 0.057284 .
AvgLessThanMin                        -2.611e-01  6.403e-02  -4.078 4.55e-05 ***
MaxCashTransfer                       -4.044e-01  5.157e-02  -7.842 4.43e-15 ***
MaxCashWithdrawal                     -2.585e-01  1.346e-01  -1.921 0.054758 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27809  on 27580  degrees of freedom
Residual deviance: 24403  on 27511  degrees of freedom
AIC: 24543

Number of Fisher Scoring iterations: 10
```

ANOVA for Full- and Reduced Model for `data.old`

```
Analysis of Deviance Table

Model 1: Recovery ~ DebtCollectionCompany + Product + Bank + SerialN +
    AgeOfAccount + AgeOfCustomer + RecruitmentChannel + MTPCollectionWarning +
    ClosingBalanceCollectionWarning + SumCollectionWarning +
    SumDunning + Recurring + SumCreditIncrease + ‘PaymentMethod ‘ +
    DebtAppl + GrossIncome + EvaluationMethod + Active + RR.avg +
    AvgClosingBalance + AvgNormal + AvgInterest + AvgPayN + AvgOverLimit +
    DaysSinceLastTime + CreditLimit + AvgCashWithdrawal + AvgLessThanMin +
    MaxCashTransfer + MaxCashWithdrawal
Model 2: Recovery ~ DebtCollectionCompany + Product + Bank + SerialN +
    AgeOfAccount + Gender + AgeOfCustomer + RecruitmentChannel +
    MTPCollectionWarning + ClosingBalanceCollectionWarning +
    SumCollectionWarning + SumDunning + Recurring + SumCreditIncrease +
    ‘PaymentMethod ‘ + DebtAppl + GrossIncomeAppl + GrossIncome +
    EvaluationMethod + Active + RR + RR.avg + AvgClosingBalance +
    AvgNormal + AvgCashTransfer + AvgPurchase + AvgInterest +
    AvgPayN + AvgOverLimit + MaxClosingBalance + DaysSinceLastTime +
    CreditLimit + AvgCashWithdrawal + AvgLessThanMin + MaxCashTransfer +
    MaxCashWithdrawal
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1     27511      24402
```

```
2     27505      24399  6   3.8598   0.6956
```

Hosmer-Lemeshow for `data.old`

```
        Hosmer and Lemeshow goodness of fit (GOF) test

data:  reduced.model$y, fitted(reduced.model)
X-squared = 3992, df = 2794, p-value < 2.2e-16
```

Regularized Model for `data.old`

```
Call:
glm(formula = Recovery ~ ., family = "binomial", data = reg.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8601   0.3177   0.5068   0.6890   3.1735

Coefficients:
                                     Estimate Std. Error z value Pr(>|z|)
(Intercept)                         1.757e+00  1.112e-01  15.797  < 2e-16 ***
ProductSpareBank.1.MasterCard.Gold  3.433e-01  3.884e-02   8.838  < 2e-16 ***
SerialN10                          -7.453e-01  8.132e-02  -9.165  < 2e-16 ***
SerialN20                          -4.204e-01  4.083e-02 -10.295  < 2e-16 ***
SerialN40                           1.183e-01  3.829e-02   3.091 0.001994 **
MTPCollectionWarning               -7.268e-05  7.471e-06  -9.728  < 2e-16 ***
SumDunning                          1.223e-01  7.262e-03  16.840  < 2e-16 ***
SumCreditIncrease                  -7.008e-02  2.417e-02  -2.900 0.003734 **
PaymentTypePrint                    3.054e-01  3.744e-02   8.157 3.43e-16 ***
EvaluationMethodOrdinary           -4.083e-01  3.751e-02 -10.885  < 2e-16 ***
RR.avg                              2.168e+00  2.060e-01  10.525  < 2e-16 ***
AvgNormal                          -8.268e-01  1.219e-01  -6.785 1.16e-11 ***
AvgCashTransfer                    -1.467e+00  2.009e-01  -7.304 2.79e-13 ***
AvgInterest                        -1.662e-04  4.868e-05  -3.415 0.000639 ***
AvgOverLimit                       -9.994e-01  4.724e-02 -21.155  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27809  on 27580  degrees of freedom
Residual deviance: 24938  on 27566  degrees of freedom
AIC: 24968

Number of Fisher Scoring iterations: 5
```

Fitting a 2^2 factorial design with log(RSS) of the SVM as response for `data.new`

```
Call:
lm.default(formula = log(RSS) ˜ ., data = X[1:4, c(1, 2, 3, 6)])

Residuals:
ALL 4 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -7.2159         NA      NA       NA
c             0.2833         NA      NA       NA
sigma         0.2552         NA      NA       NA
cs            0.1933         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 3 and 0 DF,  p-value: NA
```

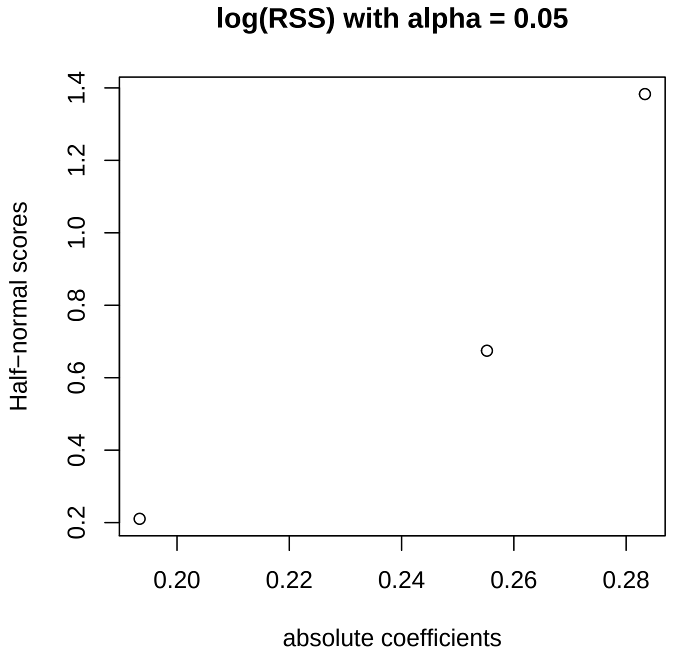


Figure B.1: Half-normal plot of the 2^2 experiment on log(RSS) for `data.new`

Fitting a 2^{4-1} factorial design with the validity indices V_{PE}, V_{PC}, V_{XB} of the PFCM model as response for `data.new`

```
Call:
lm.default(formula = pe ˜ a + b + m + eta + a:b + a:m + b:m,
    data = cbind(plan, pe))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.579227         NA      NA       NA
a1           0.009271         NA      NA       NA
b1          -0.078577         NA      NA       NA
m1           0.002711         NA      NA       NA
eta1         0.287176         NA      NA       NA
a1:b1        0.002880         NA      NA       NA
a1:m1        0.015987         NA      NA       NA
b1:m1        0.006002         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA

Call:
lm.default(formula = pc ˜ a + b + m + eta + a:b + a:m + b:m,
    data = cbind(plan, pc))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.576717         NA      NA       NA
a1          -0.020085         NA      NA       NA
b1          -0.155625         NA      NA       NA
m1          -0.025576         NA      NA       NA
eta1        -0.173290         NA      NA       NA
a1:b1       -0.003988         NA      NA       NA
a1:m1        0.045386         NA      NA       NA
b1:m1       -0.014830         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA
```

```
Call:
lm.default(formula = xb ˜ a + b + m + eta + a:b + a:m + b:m,
    data = cbind(plan, xb))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   168819         NA      NA       NA
a1            -41521         NA      NA       NA
b1            -62819         NA      NA       NA
m1             75215         NA      NA       NA
eta1           47939         NA      NA       NA
a1:b1          73894         NA      NA       NA
a1:m1         -71544         NA      NA       NA
b1:m1         -75887         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA
```

Fitting a 2^{4-1} factorial design with the validity indices V_{PE}, V_{PC}, V_{XB} of the PFCM model as response for `data.old`

```
Call:
lm.default(formula = PE ˜ a + b + m + eta + a:b + a:m + b:m,
    data = cbind(plan, PE))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.345375         NA      NA       NA
a1          0.015625         NA      NA       NA
b1          0.009625         NA      NA       NA
m1          0.015125         NA      NA       NA
eta1        0.216625         NA      NA       NA
a1:b1       0.018875         NA      NA       NA
a1:m1       0.009875         NA      NA       NA
b1:m1       0.006875         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
```

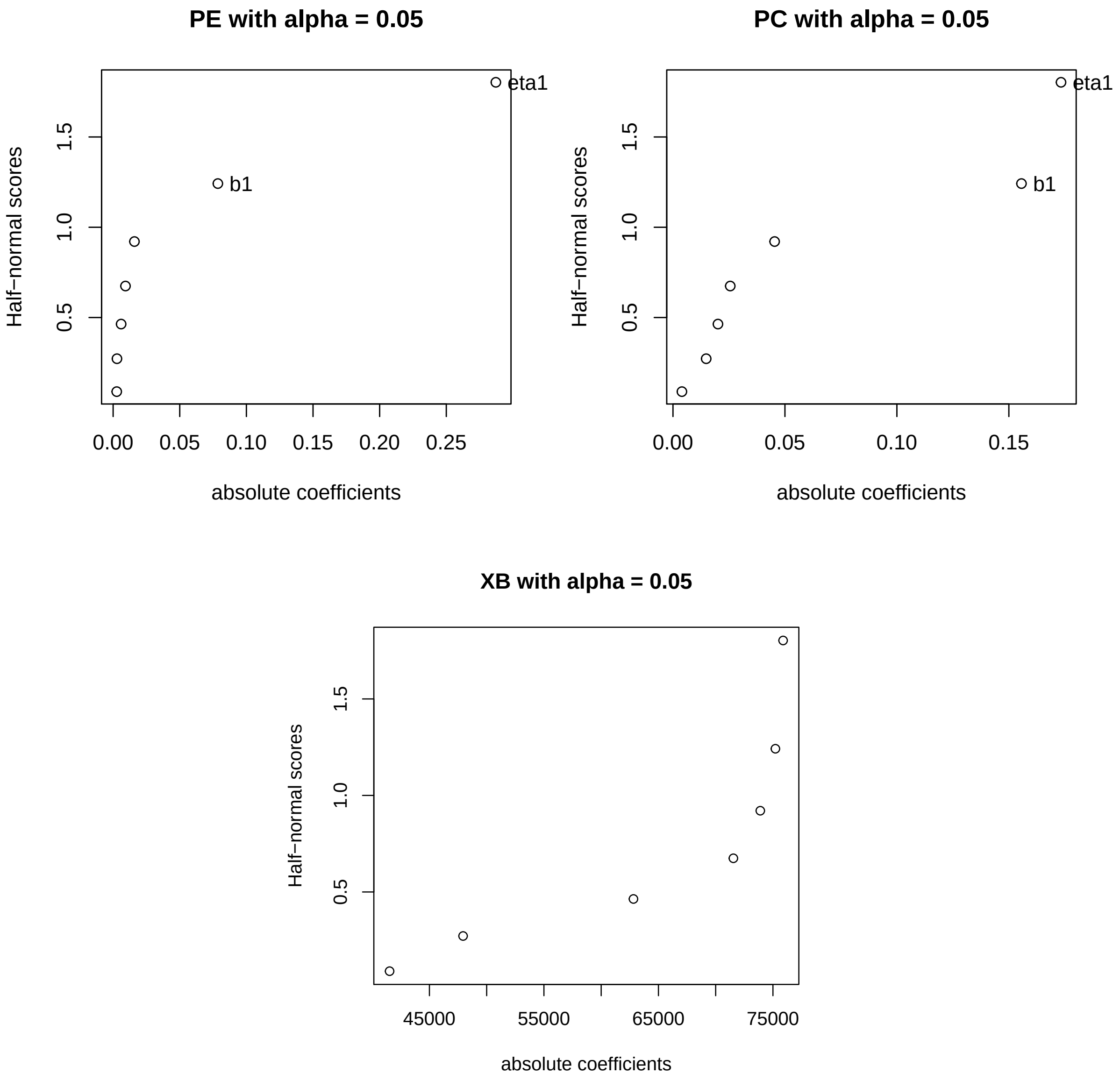


Figure B.2: Half-normal plot of the 2^{4-1} experiment on the PE, PC and XB for `data.new`

```
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA

Call:
lm.default(formula = PC ~ a + b + m + eta + a:b + a:m + b:m,
    data = cbind(plan, PC))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  0.90525         NA      NA       NA
a1           0.01050         NA      NA       NA
b1          -0.29600         NA      NA       NA
m1          -0.01025         NA      NA       NA
eta1        -0.27025         NA      NA       NA
a1:b1       -0.00225         NA      NA       NA
a1:m1        0.12300         NA      NA       NA
b1:m1        0.00600         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA

Call:
lm.default(formula = XB ~ a + b + m + eta + a:b + a:m + b:m,
    data = cbind(plan, XB))

Residuals:
ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   855174         NA      NA       NA
a1            287410         NA      NA       NA
b1           -521362         NA      NA       NA
m1             25091         NA      NA       NA
eta1         -206664         NA      NA       NA
a1:b1        -125420         NA      NA       NA
a1:m1         358158         NA      NA       NA
b1:m1         135894         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,        Adjusted R-squared:    NaN
F-statistic:   NaN on 7 and 0 DF,  p-value: NA
```

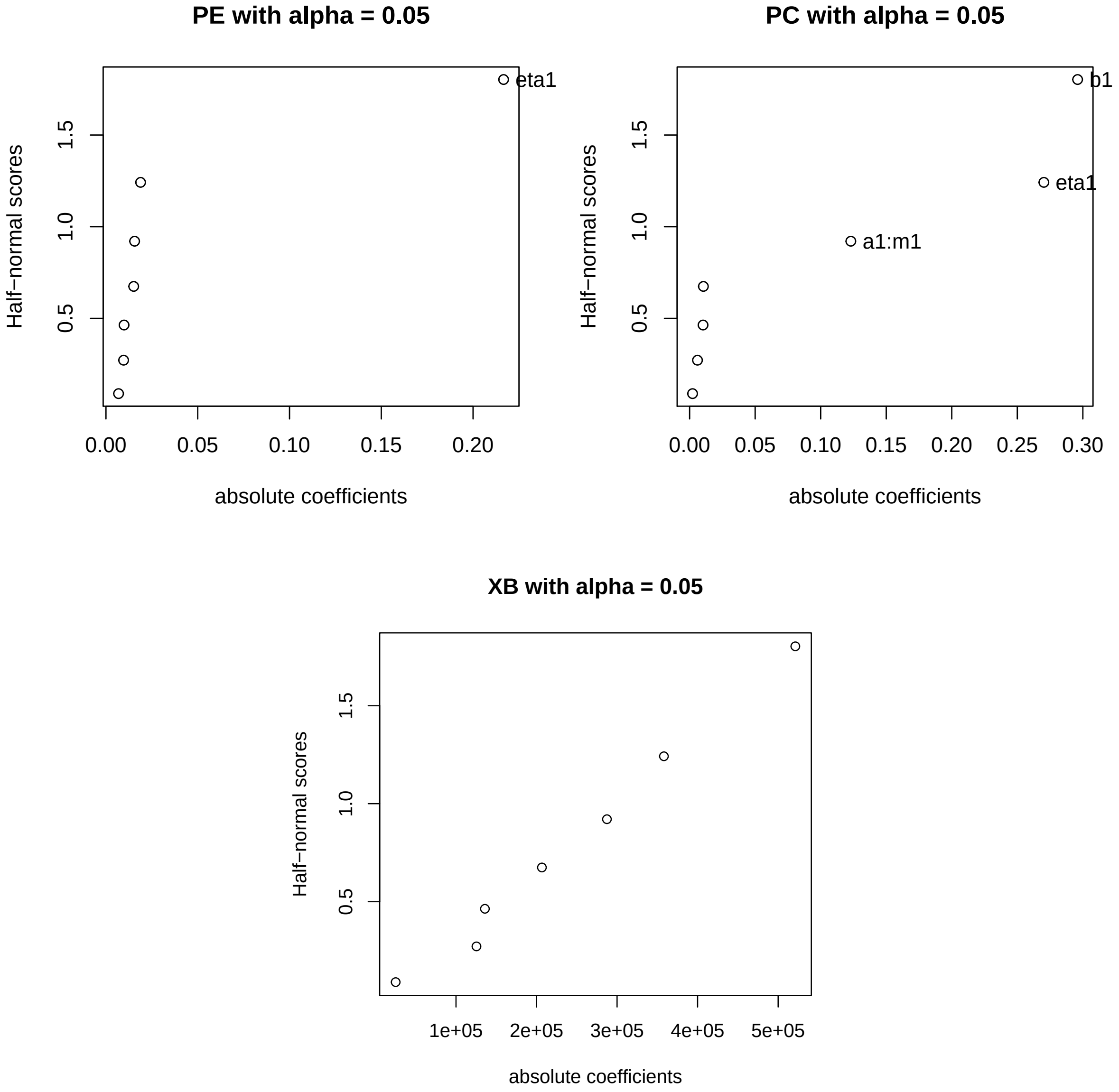


Figure B.3: Half-normal plot of the 2^{4-1} experiment on the PE, PC and XB for `data.old`