Master's thesis

2019

Sindre Nybakk Uthus

Master's thesis

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

NTNU
Norwegian University of Science and Technology

Sindre Nybakk Uthus

Binary Classification, Logistic and Nominal Regression: Application to Bank Customer Loyalty Data

June 2019

NTNU
Norwegian University of Science and Technology

Norwegian University of Science and Technology

Binary Classification, Logistic and Nominal Regression: Application to Bank Customer Loyalty Data

Sindre Nybakk Uthus

Master of Science in Physics and Mathematics
Submission date: June 2019
Supervisor: Øyvind Bakke

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

Customers are the foundation of any business's success, and a business can never be too grateful for loyal customers. Customer insight is therefore an important key to help sustain loyal and active customers. In this thesis we are going to detect significant differences between different customer types, as well as indicating future inactive customers. Doing so, we will get useful insight about the inactive customers, and perhaps understand why they choose to go from being active to being inactive. The analyses are done on a bank customer database, provided by the bank itself.

The bank customers are divided into six groups, or categories, based on customer activity and the number products used. The categories are denoted by A–F, where A contains the most active customers and F contains the least active customers with no products used. We perform nominal regression in order to detect differences between these groups. We experienced that customers that have applied for loan/credit card are more likely to be customers from the categories A and B. Furthermore, we experienced that probability for being in category F is strongly decreased if the customer is a member or a non-member with member benefits. Also, the probability for being in category A is significantly increased if the customers have activated electronical billing.

Let the categories A–D relate to the active customers, and the categories E–F relate to the inactive customers. To indicate customers that are going to be inactive in the future, we create an indicator model. We perform statistical modelling and learning methods, such as binary logistic regression, random forests and XGBoost, in order to create this model. We use model selection methods, such as the Akaike Information Criterion (AIC), lasso regularization and variable importance. The performance of a model is evaluated on the AUC value on test data. The model that performed best was the XGBoost model with all the variables included. Thus, this will be used as the indicator for detecting bank customers that are going to be inactive within the next year. We experienced that balance of the customer was clearly the most significant variable, when it comes to being active or inactive in the future. The binary logistic regression coefficient for this variable is negative. Hence, the higher the balance on the deposit account of a customer, the lower is the probability for being inactive in the future. The number of transactions of the customer and if the customer has a loan, are both also very important factors when it comes to being active/inactive in the future.

Sammendrag

Kunder er grunnlaget for enhver forretning sin suksess, og en forretning kan aldri være for takknemlig for sine trofaste kunder. I denne masteroppgaven skal vi finne signifikante forskjeller mellom ulike typer kunder. Ved å gjøre dette kan vi få nyttig kundeinnsikt av passive kunder, og forstå hvorfor kunder velger å gå fra å være aktive til å bli inaktive kunder. Analysene er gjort på kundedatabasen til banken som denne oppgaven er skrevet for.

Bankkundene er delt inn i seks grupper, eller kategorier, basert på kundeaktivitet og antall bankprodukter tatt i bruk. Vi betegner kategoriene som A–F, hvor kategori A står for kundene som er mest aktive og kategori F står for kundene som er minst aktive. Vi bruker nominal regresjon til å finne forskjeller i hver av disse kategoriene. Vi fant ut at kunder som har vært i en søkeprosess for lån eller kredittkort hadde en større sannsynlighet for å være en kunde av kategoriene A og B. Vi fant også ut at sannssynligheten for å være kategorisert i F er sterkt redusert hvis kunden er medlem eller ikke-medlem med medlemsbetingelser. I tillegg, sannsynligheten for å være en kunde av kategori A økes kraftig dersom kunden har aktivert eFaktura.

La kategoriene A–D være kategoriene med aktive kunder, og kategoriene E–F være kategoriene med inaktive kunder. Vi lager en indikatormodell som forutser hvilke kunder som går fra å være aktive til å bli inaktive i fremtiden. Vi utfører statistiske modellerings- og læringsmetoder, som binær logistisk regresjon, “random forests” og XGBoost, for å oppdage disse kundene. Vi tester også forskjellige modeller basert på modellseleksjonsmetoder som bruker Akaike informasjonskriterium (AIC), lasso regularisering og viktighet av variabler. Modellkvaliteten er evaluert på AUC-verdien på test data. Modellen som presterte best var XGBoost-modellen med alle variablene inkludert. Dermed vil denne modellen bli brukt som indikatormodellen, m.a.o. modellen som skal forutse om en aktiv bankkunde holder seg aktiv eller blir inaktiv i løpet av det kommende året. Vi erfarte at innskuddssaldoen til kunden var den mest signifikante variabelen i denne analysen. Den logistiske regresjonskoeffisienten for denne variabelen er negativ, som betyr at jo høyere innskuddsaldoen er, desto lavere sannsynlighet er det for å bli inaktiv i fremtiden. Antall transaksjoner av kunden og om kunden har lån, er også svært viktige faktorer som spiller inn.

Preface

This master's thesis concludes my M.Sc. degree in Industrial Mathematics at Norwegian University of Science and Technology. The project was done in cooperation with, and written for, an anonymous, Norwegian bank. The thesis was written during the spring 2019 semester, with Øyvind Bakke as supervisor.

I want to thank my supervisor Øyvind Bakke for getting me in touch with a real-world project and marvelous guiding throughout the project. I would also like to thank the business analyst team of the respective bank for the data sets used in the project and other valuable input.

Abbreviations

c	=	The prediction value of a leaf node in a decision tree
d	=	The number of categories, excluding a reference category
i	=	Customer index
j	=	Variable index
k	=	The number of explanatory variables (predictors)
m	=	Decision tree/basis function index
n	=	The number of customers
p	=	The total number of variables, equal to $k + 1$
$\hat{p}_{tr}$	=	Proportion of class r in region R_t
r	=	Category index
s	=	Splitting value in decision trees
t	=	Leaf node index
u	=	A latent utility model
w	=	Weight parameter
x	=	A single data value from X
$\mathbf{x}_i$	=	The data vector for a observation i
y	=	A response variable
$\mathbf{y}$	=	A response vector
z	=	The size of the column subset P_0
C	=	A cumulative distribution function
D	=	Deviance, or cross entropy
E	=	Misclassification error
F	=	The expected Fisher information matrix
G	=	Gini index
H	=	The observed Fisher information matrix
I	=	Identity matrix or identity function
K	=	The number of categories, equal to $d + 1$
L	=	Likelihood function
M	=	The total number of trees
R	=	The predictor regions in a decision tree
T	=	The number of leaf nodes in a decision tree
W	=	A weight vector
X	=	A data matrix

α	=	Some positive constant used in loss functions
β	=	Regression coefficient
γ	=	Penalizing parameter for T
δ_c	=	A column subsampling fraction
δ_r	=	A row subsampling fraction
ϵ	=	A positive value sufficiently close to zero
ε	=	An error term
ζ	=	A dispersion parameter
η	=	The linear predictor
θ	=	A parameter of interest
κ	=	The number of levels of a categorized variable
λ	=	Penalizing parameter used in lasso regularization
μ	=	Mean value
ν	=	Weight-smoothing parameter for w in XGBoost
π	=	Probability or proportion
ρ	=	Correlation parameter
σ	=	Standard deviance
τ	=	Parameter between 0 and 1, related ROC-curves
ϕ	=	Decision tree (basis function in XGBoost)
Γ_{split}	=	The maximum gain at node splits in ϕ
Σ	=	Covariance matrix
Ω	=	Complexity penalizing term
$\mathcal{L}$	=	Loss function
AIC	=	Akaike Information Criterion
AUC	=	Area under curve, (for ROC-curves)
GLM	=	Generalized linear models
LOOCV	=	Leave-one-out cross-validation
MLR	=	Multiple linear regression
MSE	=	Mean squared error
ROC	=	Receiver operating characteristic
RSS	=	Residual sum of squares

Table of Contents

1 Introduction and Problem Description

How many customers a bank has, does all depend on how a customer is defined. The last 8–9 years, the respective bank has classified a customer as either *active* or *inactive*, based on whether the customer has created a deposit and/or a lending product or not. The *old customer categorization* can be seen in Table 1. A problem with this categorization is that it is too general, and thereby superficial. Almost half of the active customers don't have capital transactions registered the last year. This means they are not as active as the category suggests.

The bank has a new proposal of how to categorize the customers. The customers are classified into six categories A–F. We denote category A, B, C and D as the desired customers and category E, F as potensial customers. The *new customer categorization* can be seen in Table 2. Given the new customer categorization we want to investigate the following:

- Are there any significant differences between the customers in each category?
- Develop an indicator that indicate if an active customer (category A, B, C, D) soon will be inactive (category E, F).

For the first and smaller task, nominal regression can be used to see the effects of each variables within different groups. Nominal regression models the linear predictor $\eta_{ir} = \ln(\pi_{ir}/\pi_{i,c+1})$, for observation i and category r. Category $c + 1$ is the reference category, which in this case will be category A. Looking at the regression coefficient estimates, we will se the change in probability for the customer to be in category r, relative to category A.

For the second and bigger task, we follow a statistical learning approach to develop the indicator. We look at learning approaches such as random forests and XGBoost. Binary logistic regression is also used, but mainly as a simple guide to interpret the effects of different variables on the response variable. It is also used for observing the bad prediction accuracy relative to tree-based ensemble methods.

Category	Description
Active	Customer with deposit and/or lending products.
Inactive	No deposit and/or lending products.

Table 1: Categorization of customers currently used by the bank.

Category	Description	Criteria
A	Full customer	Payroll input and several products
B	Customer with several products	Several products, deposit customers must have capital transactions within the last year.
C	Customer with one product	Only one product, deposit customers must have capital transactions within the last year, loan co-signers do not count.
D	Savings customer	Deposit customers with no capital transactions within the last year, but with balance over NOK 1000.
E	Inactive customer	Customer with balance equal to zero or deposit balance under NOK 1000. In addition, the customer does not have a credit card nor capital transactions within the last year.
F	Passive customer	No deposit and/or lending products.

Table 2: Proposal of new categorization of customers.

2 Theory

Here we will describe the theory that we will use in practice later in this thesis. In Section 2.1 we look at some background theory. This includes multiple linear regression (MLR), generalized linear methods (GLM) and decision trees. Further, we present the theory from two special cases of GLM, namely logistic regression and nominal regression, described in Section 2.2 and 2.3, respectively. In Section 2.4 we introduce some variable selection and regularization methods, such as AIC and lasso regularization. The theory behind random forests and XGBoost are presented in Section 2.5, which will be used to create the indicator model. Finally, we will use ROC plots and AUC, whose theory are described in Section 2.6, to evaluate the model performance. Most of the theory is taken from Fahrmeir et al. (2013) and James et al. (2013).

2.1 Background theory

We start off with by describing the theory of the essentials within statistical analysis and modelling. This theory will be the foundation of the theory that comes after, starting at Section 2.2.

2.1.1 Multiple Linear Regression

In multiple linear regression, or MLR, the response variable, $\mathbf{y}$, is described by

$$\mathbf{y} = X\boldsymbol{\beta} + \varepsilon,$$

where X is a data matrix, $\boldsymbol{\beta}$ is a vector of coefficients and ε is a vector of error terms. The ε is Gaussian distributed with mean zero and variance σ^2, i.e, $\varepsilon \sim N(0, \sigma^2 I)$, where I is the identity matrix. Denote by n the number of observations in the data set, and k the number of variables. The vector of coefficients also includes the intercept, β_0, so we denote $p = k+1$ as the number of coefficients we want to estimate. Then X will be a $n \times p$ matrix, where the first column corresponding to β_0, which only contains ones, and $\boldsymbol{\beta}$ will be a $p \times 1$ vector containing the regression coefficients. Let $\mathbf{x}_i$ and y_i be the variable values and the response value for observation i, respectively. We assume the pairs $(\mathbf{x}_i, y_i)$ are independent of each other, and also that X has full rank. Then y_i follows a Gaussian distribution with mean

$$\mathrm{E}(y_i) = \mathbf{x}_i^T \boldsymbol{\beta},$$

and variance,

$$\mathrm{Var}(y_i) = \sigma^2.$$

So how is $\boldsymbol{\beta}$ estimated? There are two different ways to estimate $\boldsymbol{\beta}$, least squares and maximum likelihood estimator. The least squares estimate can be found by minimizing the RSS (residual sum of squares), $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where $\hat{y}_i = \beta_0 + \sum_{j=1}^{k} x_{ij}\beta_j$. Since this is a convex function of $\boldsymbol{\beta}$, it follows that it has some minimum value. A perfect regression line would result in a minimum value of zero. The least squares method can also be written on matrix form,

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y}.$$

The maximum likelihood estimator can be found by likelihood theory. We will come back to this in the next section. Both of the methods give the same estimates for $\boldsymbol{\beta}$.

2.1.2 Generalized Linear Models

The linear regression model is well suited for regression analyses where the response variable is continuous and somewhat normally distributed. For cases when the response is not normally distributed, we need an appropriate transformation of the response. Let us assume that the response is binary, i.e. the $y_i \in \{0, 1\}$. The response in linear models can take any value between $-\infty$ and ∞. Therefore, we need to somehow transform the response values into values between 0 and 1. We introduce *generalized linear models* (GLM), to investigate the more general cases of linear modelling. GLM contains three important components (Fahrmeir et al., 2013):

1. The random component: The distribution of the response variable y_i.

2. The systematic component: The linear predictor $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$, which is simply a linear combination of the explanatory variables.

3. The link function: The linear predictor is connected to the mean of the response $\mu_i = \mathrm{E}(y_i)$ by a link function, $\mu_i = g(\eta_i)$. The inverse of the link function is called the response function, and is denoted as $h(\mu_i)$.

For regular MLR, the random component is simply $y_i \sim N(\mu_i, \sigma^2)$. The response is normally distributed with mean μ_i and nuisance parameter σ^2. The systematic component is $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$. The link function is $g(\eta_i) = \eta_i$, which means that the linear predictor is the estimated mean of the distribution. This is called the identity link. The three components for the most common distributions are given in Table 3.

If the response distribution belongs to an exponential family, then the distribution has some useful properties. The exponential families include some of the most important distributions, such as binomial, poisson and categorical. Since all of these distributions originate from the exponential families, we can use the same algorithm for all of them. A univariate exponential family has a pdf (or pmf) of the form

$$f(y_i \mid \theta_i) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\zeta} \cdot w_i + c(y_i, \zeta, w_i)\right). \tag{2.1}$$

where ζ is a dispersion parameter, or nuisance, θ_i is the parameter of interest and w_i some weight. The functions b and c are specific for each exponential

	R.C.	S.C.	Link function
Gaussian regression (MLR)	$N(\mu_i, \sigma^2)$	$\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$	$\eta_i = \mu_i$
Exponential regression	$\text{Exp}(\lambda_i)$	$\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$	$\eta_i = -\frac{1}{\mu_i}$
Poisson regression	$\text{Poiss}(\lambda_i)$	$\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$	$\eta_i = \ln \mu_i$
Binary regression	$\text{Bin}(1, \pi_i)$	$\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$	$\eta_i = \ln \frac{\mu_i}{1-\mu_i}$

Table 3: Important components for GLM. Here, the random component (R.C.), the systematic component (S.C.) and the link function are listed for the Gaussian, exponential, poisson and bernoulli distribution.

family. The mean and variance of this distribution can be found by

$$\text{E}(y_i) = b'(\theta_i) \quad \text{and} \quad \text{Var}(y_i) = b''(\theta_i) \cdot \frac{\zeta}{w_i},$$

respectively (Fahrmeir et al., 2013). Therefore, we need b to be twice differentiable. The GLM framework also requires b to be a one-to-one function.

The most important goal in generalized linear models, including in multiple linear models, is to estimate the regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$ and their significance. We introduce the likelihood as the the product of the densities of every observation y_i,

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

where $L_i(\boldsymbol{\beta})$ are each of the n individual likelihood contributions. We want to maximize the likelihood $L(\boldsymbol{\beta})$, or the log-likelihood $\ln L(\boldsymbol{\beta})$, through the unknown parameter $\boldsymbol{\beta}$. This is known as *maximum likelihood estimation* (Fahrmeir et al., 2013). The log-likelihood is simply

$$l(\boldsymbol{\beta}) = \ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ln L_i(\boldsymbol{\beta}).$$

We want to maximize this function, hence the first derivative of the log-likelihood with respect to $\boldsymbol{\beta}$ is needed. This is called the score function, and it can be shown that

$$s(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} s_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\partial l_i(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)\mathbf{x}_i h'(\eta_i)}{\text{Var}(y_i)}.$$

This can also be written on matrix form, $s(\boldsymbol{\beta}) = X^T D \Sigma^{-1}(\mathbf{y} - \mu)$, where $D = \text{diag}(h'(\eta_i))$ and $\Sigma = \text{diag}(\text{Var}(y_i))$. In order to maximize the log-likelihood, we require that $s(\hat{\boldsymbol{\beta}}) = 0$. This results in the *ML equations* (Fahrmeir et al., 2013), which is a set of p equations. These equations can either be solved analytically or iteratively, depending on the equations are linear or not. The Newton–Raphson and Fisher scoring are the most popular algorithms used to solve non-linear ML equations. These algorithms require either the observed or the expected Fisher information matrix. Denote by $H(\boldsymbol{\beta})$ and $F(\boldsymbol{\beta})$ the observed and expected Fisher information respectively, then

$$H(\boldsymbol{\beta}) = -\frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^T} \quad \text{and} \quad F(\boldsymbol{\beta}) = \text{Cov}(s(\boldsymbol{\beta})).$$

Furthermore, these matrices are connected through $F(\boldsymbol{\beta}) = E(H(\boldsymbol{\beta}))$ (Fahrmeir et al., 2013). The expected Fisher information can also be written on matrix form, $F(\boldsymbol{\beta}) = X^T W X$, where $W = \text{diag}(h'(\eta_i)^2/\text{Var}(Y_i))$.

The *Newton–Raphson* algorithm is defined by

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \hat{\boldsymbol{\beta}}^{(i)} - \frac{s\left(\hat{\boldsymbol{\beta}}^{(i)}\right)}{s'\left(\hat{\boldsymbol{\beta}}^{(i)}\right)},$$

where i is the current iteration. The derivative of the score function $s'(\boldsymbol{\beta})$ is the negative of the observed Fisher information $F(\boldsymbol{\beta})$, i.e. $s'(\boldsymbol{\beta}) = -H(\boldsymbol{\beta})$, hence it can be rewritten as

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \hat{\boldsymbol{\beta}}^{(i)} + H^{-1}\left(\hat{\boldsymbol{\beta}}^{(i)}\right) \cdot s\left(\hat{\boldsymbol{\beta}}^{(i)}\right)$$

The *Fisher scoring* method is obtained when the observed Fisher information is replaced by the expected Fisher information,

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \hat{\boldsymbol{\beta}}^{(i)} + F^{-1}\left(\hat{\boldsymbol{\beta}}^{(i)}\right) \cdot s\left(\hat{\boldsymbol{\beta}}^{(i)}\right).$$

Since we make use of $F^{-1}(\boldsymbol{\beta})$, the data matrix X needs to be full rank. This should be no problem if $n > p$. For GLM, this can also be written on matrix form as an iteratively reweighted least squares (IRLS),

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \left(X^T W\left(\hat{\boldsymbol{\beta}}^{(i)}\right) X\right)^{-1} X^T W\left(\hat{\boldsymbol{\beta}}^{(i)}\right) Y^{(i)},$$

where W are weights defined as before. In order to start the algorithm, an initial value $\hat{\boldsymbol{\beta}}^{(0)}$ is required. The iterations stop when the algorithm has met some convergence criterion, for example when $\left\|\hat{\boldsymbol{\beta}}^{(i+1)} - \hat{\boldsymbol{\beta}}^{(i)}\right\| / \left\|\hat{\boldsymbol{\beta}}^{(i)}\right\|$ is smaller than some positive value ϵ. Then we set $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{(i)}$, because the algorithm has reached a steady state.

2.1.3 Decision Trees

Tree–based methods are simple methods used for modelling and prediction of data. The simplest model within tree-based methods is called a *decision tree*. The main idea behind building a tree is to split up the data into different regions, based on a set of decision rules. In order to make prediction for a new observation, we classify the observation into one of these regions by applying the decision rules.

A single decision tree is drawn as an upside down tree, where the top node is called the *root*. The bottom nodes with no splitting criteria are called terminal nodes, or *leaf nodes*. Each terminal node represents one of the predictor regions, $R = \{R_1, R_2, \dots, R_T\}$, where T is the number of leaf nodes. The points along the tree where the predictor space is split, are referred to as *internal nodes*. Denote by X the data matrix and by $f(X)$ the predicted values of X. Then

$$f(X) = \sum_{t=1}^{T} c_t \cdot I(X \in R_t), \tag{2.2}$$

where c_t is some prediction constant and $I(X \in R_t)$ is the indicator function. If $X \in R_t$, the indicator function will return one, otherwise it will return zero. Applying the decision rules to X, we assign the value c_t to the observations in X that fall into region R_t. A single decision tree is easy to interpret, but not necessarily a good way to predict data. Figure 1 illustrates a single tree structure. We can split decision trees into two groups, regression trees and classification trees.

Regression trees are trees where the response is quantitative. For every observation that falls into region R_t, we make the same prediction, the mean of all response values in the predictor space R_t. To obtain the regions in R, we need to make some splitting criterion that minimizes the prediction error. For regression trees, a top-down, greedy algorithm called *recursive binary splitting* is used. This is done by starting with all of the observations in a single region of the predictor space. Then we split this region into two new regions, R_1 and R_2. The splitting criterion will split the region based on the predictor x_j and a splitting value s, such that $R_1(j,s) = \{x \mid x_j < s\}$ and $R_2(j,s) = \{x \mid x_j \geq s\}$. The quantity we want to minimize is then

$$\sum_{i: x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2. \tag{2.3}$$

Doing so, we have created to new regions. Furthermore, we want to split these again and again, until we reach some stopping criterion. The stopping criterion can for example be when the reduction in RSS $= \sum_i (y_i - \hat{y}_i)^2$ is smaller than some limit, or there is less than a minimum number of observations left in a certain region.

Classification trees are trees where the responses are categories. Denote by K the number of categories. Often we have only two categories, i.e. $K = 2$, which is called binary classification. For every observation that falls into region R_t, we make the same prediction, the category with most responses in the predictor space R_t. To obtain the regions R, we need to make some splitting criterion that minimizes the prediction error. This approach will be different than for regression trees. One of the easiest methods of measuring error in a classification

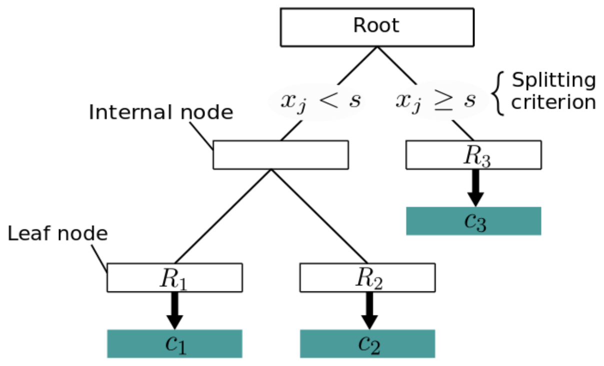


Figure 1: This is a visualization of a single tree structure. The root node is split based on predictor x_j and splitting value s. Here, the predictor space R are divided into $T = 3$ regions. An observation that falls into region R_t will be assigned the predicted value c_t.

tree is the *misclassification rate*. We observe the number of misclassifications in region R_t, and simply divide it by N_t, the number of observations in this region. The following error measurements are taken from James et al. (2013). Denote by

$$\hat{p}_{tr} = \frac{1}{N_t} \sum_{x_i \in R_t} I(y_i = r),$$

the proportion of class r in region R_t. Then the misclassification rate is

$$E = 1 - \max_k(\hat{p}_{tr}),$$

which is the proportion of failed classifications. We also have the *Gini index*, which is defined by

$$G = \sum_{r=1}^{K} \hat{p}_{tr}(1 - \hat{p}_{tr}).$$

The Gini index measures the variance within each class k, and then computes the total sum of these. We observe that if $\hat{p}_{tr}$ is close to zero or one, then $G \approx 0$. Therefore, G will be some measurement of the so-called impurity in region R_t. An alternative to the Gini index is *cross-entropy*, or *deviance*. The cross-entropy is defined by

$$D = -\sum_{r=1}^{K} \hat{p}_{tr} \log \hat{p}_{tr}. \tag{2.4}$$

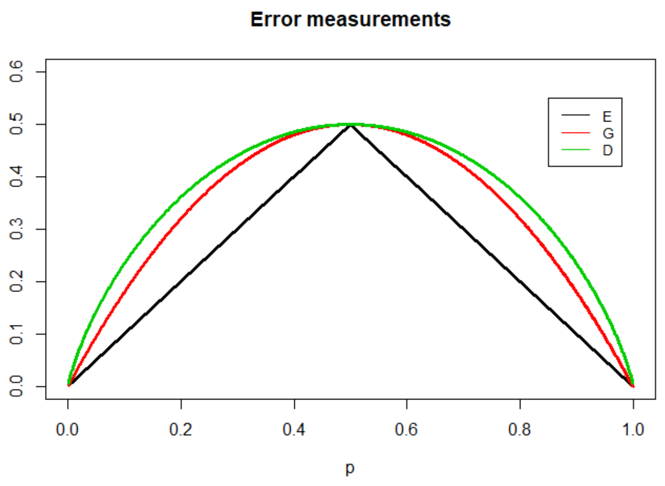


Figure 2: This illustrates three different error measurements for binary classification trees, namely misclassification rate E, Gini index G and deviance D. Deviance is scaled to go through point $(0.5, 0.5)$.

Again, D will be close to zero, if $\hat{p}_{tk}$ are close to zero or one. For a binary classification problem, i.e. $K = 2$, we only have two probabilities, π_1 and π_2. We can use the fact that $\pi_2 = 1-\pi_1$ and get $E = 1-\max(\pi, 1-\pi)$, $G = 2\pi(1-\pi)$ and $D = -\pi\log(\pi)-(1-\pi)\log(1-\pi)$ (Friedman et al., 2001). The classification error measurements are shown in Figure 2.

2.1.4 k-Fold Cross-Validation

Cross-validation is an approach to evaluate a model on data set. The evaluation is based on an average error rate. Define *training set* as the data set on which the model is trained and *test set* as the data set on which the model will be evaluated. The data set is divided into a single training set and a single test set. It is common to assign the majority of the data to the training set, for example 80% of the original data set, and assign the remaining data to the test set. A test set is necessary to see how the model performs on new observations.

The k-fold cross-validation approach is simple. The training set, X, is divided into k equally sized sets (folds). For the first iteration we keep the first fold as a test fold, and train our model on the remaining $k-1$ training folds. The model is evaluated on the test set by calculating the test error, $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Denote MSE_i as the test error for iteration i. For iteration two we let the second fold be the test fold, and train our model on the remaining $k-1$ training folds.

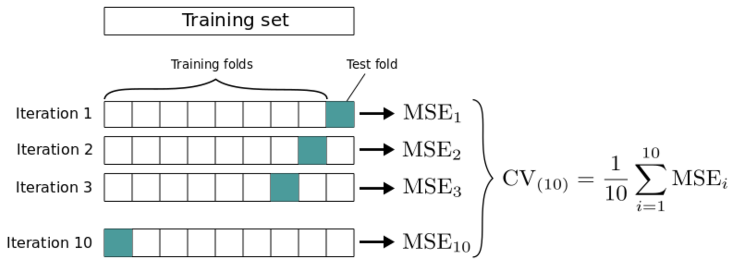


Figure 3: This is an illustration of 10-fold cross-validation. The mean squared error (MSE) is calculated at each iteration. The final cross-validation error is the average of these $k = 10$ mean squared errors.

After k iterations we calculate the average MSE, which we will define as

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

The approach is illustrated in Figure 3. The most common choices of k are 5 and 10. Having $k = n$ means we let one single observation be the test fold, and the remaining observations be the training folds. This is known as leave-one-out cross-validation (LOOCV). LOOCV has far less bias, because we fit n models that is trained on $n-1$ of the observations. However, it is very time consuming if n is really big.

2.2 Logistic Regression

Logistic regression, or binary regression, is used to model a data set where the response is binary, i.e., has two classes. We can think of the response as either success or failure. Since a binary response follows the binomial distribution, we can make use of the GLM framework which was explained in Section 2.1.2.

2.2.1 The Binomial Distribution

First let's assume that the response y_i is Bernoulli distributed, i.e. $y_i \sim \text{bin}(n_i, \pi_i)$ with $n_i = 1$. The Bernoulli distribution has probability mass function

$$f(y_i \mid \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}, \tag{2.5}$$

where y_i is either 1 or 0 and π_i is the probability that the observation is 1. The mean and variance of the Bernoulli distribution is $\mu_i = \text{E}(y_i) = \pi_i$ and

$\text{Var}(y_i) = \pi_i(1-\pi_i)$, respectively. The distribution can be rewritten in an univariate exponential form, as in (2.1),

$$f(y_i \mid \pi_i) = \exp\left(y_i \ln \frac{\pi_i}{1-\pi_i} + \ln(1-\pi_i)\right),$$

where $\phi = 1, w_i = 1, \theta_i = \ln(\pi_i/(1-\pi_i))$ and $b(\theta_i) = \ln(1+\exp\theta_i)$.

2.2.2 Canonical Link

We have the canonical link when the linear predictor is equal to the parameter θ_i, i.e. $\theta_i = \eta_i$. Letting the canonical link be the link function, some major simplifications follow. We will obtain a concave likelihood, and also the expected Fisher information matrix $F(\beta)$ will be equal to the observed Fisher information matrix $H(\beta)$. We will come back to this later. The above definition gives

$$\eta_i = g(\pi_i) = \ln \frac{\pi_i}{1-\pi_i} \tag{2.6}$$

as the canonical link in logistic regression, which is called the *logit function*. Similarly, we have the response function

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

which is basically the inverse of the logit function (see graph in Figure 4).

2.2.3 MLE of β for Logistic Regression

Given the Bernoulli distribution in (2.5), the likelihood for logistic regression is defined by

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

The log-likelihood is simply the logarithm of the likelihood, yielding

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \ln \frac{\pi_i}{1-\pi_i} - \ln(1-\pi_i). \tag{2.7}$$

Given the the logit function in (2.6) as the link function, this can also be written as

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i\eta_i + \ln\left(1+\exp\eta_i\right)\right). \tag{2.8}$$

Furthermore, taking the derivative of the log-likelihood with respect to $\boldsymbol{\beta}$, we obtain the score function. This can be done by using the chain rule $\frac{dl_i}{d\boldsymbol{\beta}} = \frac{dl_i}{d\pi_i}\frac{d\pi_i}{d\eta_i}\frac{d\eta_i}{d\boldsymbol{\beta}}$ on (2.7), or $\frac{dl_i}{d\boldsymbol{\beta}} = \frac{dl_i}{d\eta_i}\frac{d\eta_i}{d\boldsymbol{\beta}}$ on (2.8). Both rules results in the same, simple score function,

$$s(\boldsymbol{\beta}) = \sum_{i=1}^{n} s_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\partial l_i(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} = \sum_{i=1}^{n} \mathbf{x}_i(y_i - \pi_i).$$

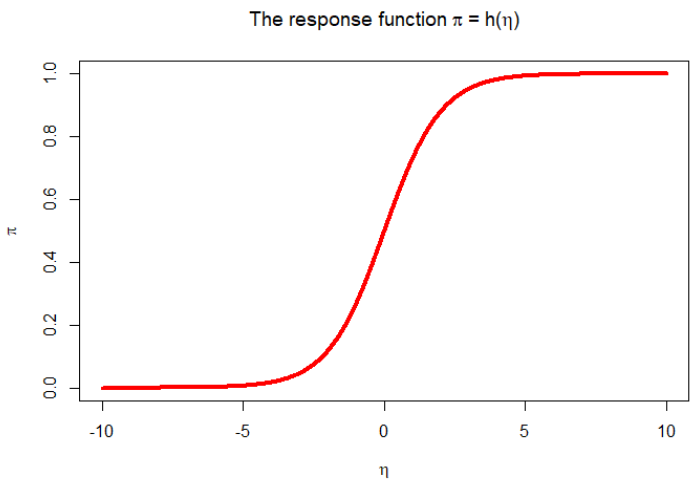


Figure 4: An illustration of the response function, also known as the inverse logit function. The more negative the linear predictor η_i is, the lower is the probability for observation i to be assigned to class 1. Similarly, the more positive the linear predictor η_i is, the higher is the probability for observation i to be assigned to class 1.

The regression coefficients can be estimated by

$$s(\hat{\boldsymbol{\beta}}) = 0, \tag{2.9}$$

and if we replace π_i with $\frac{\exp \eta_i}{1+\exp \eta_i}$, we observe that the score function gives non-linear ML equations of $\boldsymbol{\beta}$,

$$s(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \mathbf{x}_i \left(y_i - \frac{\exp \eta_i}{1 + \exp \eta_i} \right) = \sum_{i=1}^{n} \mathbf{x}_i \left(y_i - \frac{\exp(\mathbf{x}_i^T \hat{\boldsymbol{\beta}})}{1 + \exp(\mathbf{x}_i^T \hat{\boldsymbol{\beta}})} \right) = 0$$

Thus, we need to solve (2.9) iteratively. This means that the observed or expected Fishser information is required to find the MLE of $\boldsymbol{\beta}$. Let us first calculate the observed Fisher information,

$$\begin{aligned}
H(\boldsymbol{\beta}) &= -\frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^T} \\
&= -\frac{\partial}{\partial \boldsymbol{\beta}} \sum_{i=1}^{n} \left(\mathbf{x}_i(y_i - \pi_i(\boldsymbol{\beta}))\right) \\
&= \sum_{i=1}^{n} \mathbf{x}_i \frac{\partial \pi_i(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \\
&= \sum_{i=1}^{n} \mathbf{x}_i \frac{\partial \eta_i}{\partial \boldsymbol{\beta}} \frac{\partial \pi_i(\boldsymbol{\beta})}{\partial \eta_i} \\
&= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \pi_i(1 - \pi_i).
\end{aligned}$$

The observed Fisher information is used in the Newton–Raphson algorithm. Remember that $\pi_i = \pi_i(\boldsymbol{\beta})$, hence π_i, and therefore $F(\boldsymbol{\beta})$, will be updated for every iteration. Let us calculate the expected Fisher information,

$$\begin{aligned}
F(\boldsymbol{\beta}) &= \sum_{i=1}^{n} F_i(\boldsymbol{\beta}) \\
&= \sum_{i=1}^{n} \mathrm{Cov}(s_i(\boldsymbol{\beta})) \\
&= \sum_{i=1}^{n} \mathrm{E}(s_i(\boldsymbol{\beta}) s_i(\boldsymbol{\beta})^T) \\
&= \sum_{i=1}^{n} \mathrm{E}(\mathbf{x}_i \mathbf{x}_i^T (y_i - \pi_i)^2) \\
&= \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \mathrm{E}((y_i - \pi_i)^2)
\end{aligned}$$

Remember that $\mathrm{E}((y_i - \pi_i)^2)$ is simply the variance $\mathrm{Var}(y_i)$. The variance of y_i which is Bernoulli distributed is $\pi_i(1 - \pi_i)$, hence we can replace $\mathrm{E}((y_i - \pi_i)^2)$ with $\pi_i(1 - \pi_i)$,

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \pi_i(1 - \pi_i),$$

which we already know as the observed Fisher information $H(\boldsymbol{\beta})$. Since we used the canonical link function, $H(\boldsymbol{\beta})$ and $F(\boldsymbol{\beta})$ are identical. Similar to the observed Fisher information, π_i is a function of $\boldsymbol{\beta}$ which is why the expected Fisher information also is a function of $\boldsymbol{\beta}$. This results in the Fisher scoring algorithm,

$$\hat{\boldsymbol{\beta}}^{(i+1)} = \hat{\boldsymbol{\beta}}^{(i)} + F^{-1}\left(\hat{\boldsymbol{\beta}}^{(i)}\right) \cdot s\left(\hat{\boldsymbol{\beta}}^{(i)}\right)$$

which we can use to find the estimated $\boldsymbol{\beta}$.

2.2.4 Interpretation of Regression Coefficients

The regression coefficients can be hard to interpret. This is due to the logit link function. To understand what they actually mean, we introduce the *log odds*. The log odds is defined by the logarithm of the probability of success divided by the probability of failure, i.e. $\ln \frac{P(Y_i=1)}{P(Y_i=0)} = \ln \frac{\pi_i}{1-\pi_i}$. Using the response function to replace π_i, i.e. $\pi_i = h(\eta_i)$, we get

$$\ln \frac{\pi_i}{1-\pi_i} = \eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

The log odds lead us to a fairly simple expression, which we can use to interpret the β_j's. Let us see what happens when we increase the variable value x_{ij} by 1.

$$\begin{aligned}\ln \frac{P(Y_i = 1 \mid x_{ij}+1)}{P(Y_i = 0 \mid x_{ij}+1)} &= \beta_0 + \beta_1 x_{i1} + \cdots + \beta_j(x_{ij}+1) + \cdots + \beta_k x_{ik} \\ &= \beta_j + \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} \\ &= \beta_j + \ln \frac{P(Y_i = 1 \mid x_{ij})}{P(Y_i = 0 \mid x_{ij})}\end{aligned}$$

We observe that the log odds change with an additive term of β_j.

- If $\beta_j > 0$, then the log odds will increase by β_j.
- If $\beta_j < 0$, then the log odds will decrease by β_j.
- If $\beta_j = 0$, then the log odds will stay the same.

2.3 Nominal Regression

When the response is categorical, we can choose between two types of responses. The *nominal* response is chosen if the categories are unordered, i.e., the categories do not have any form of ordering to each other. Typical categories for this type of response could be types of colors or names of locations. The *ordinal* response is used when we do have ordered categories. Here we will take a closer look at nominal regression.

2.3.1 The Multinomial Distribution

The multinomial distribution is similar to the binomial distribution. Instead of two categories, consider a response having $K = d + 1$ categories. Having $d = 1$ results in logistic regression. Category $d + 1$ will then be the reference category and the rest will have their probability estimated. Denote by π_{ij} the probability that the response Y_i falls into category j. The probabilities sum to one, i.e., $\sum_{s=1}^{d+1} \pi_{is} = 1$. Thus, the last probability, $\pi_{i,d+1}$, can be calculates from the first d probabilities, i.e. $\pi_{i,d+1} = 1 - \sum_{s=1}^{d} \pi_{is}$.

The multinomial distribution for one observation has pmf

$$f(\mathbf{y} \mid \boldsymbol{\pi}) = \pi_1^{y_1} \cdots \pi_d^{y_d} (1 - \pi_1 - \cdots - \pi_d)^{1-y_1-\cdots-y_d} \tag{2.10}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_d)$, $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_d)$ and the number of independent trials is $m = 1$. If the response comes from category j, then $y_j = 1$ and the rest of $\mathbf{y}$ is 0. The distribution has the following mean and covariance matrix

$$\mathrm{E}(\mathbf{y}) = \boldsymbol{\pi} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_d \end{pmatrix},$$

$$\mathrm{Cov}(\mathbf{y}) = \begin{pmatrix} \pi_1(1-\pi_1) & -\pi_1\pi_2 & \cdots & -\pi_1\pi_d \\ -\pi_2\pi_1 & \pi_2(1-\pi_2) & \cdots & -\pi_2\pi_d \\ \vdots & \vdots & \ddots & \vdots \\ -\pi_d\pi_1 & -\pi_d\pi_2 & \cdots & \pi_d(1-\pi_d) \end{pmatrix}.$$

2.3.2 Nominal Models

Similar to logistic regression, we will now estimate the probabilities $\boldsymbol{\pi}$. Denote π_{ir} as the probability that observation i belongs to category r, where $r = 1, 2, \ldots, d+1$. This probability can be defined as

$$\pi_{ir} = P(y_i = r) = P(y_{ir} = 1).$$

The observed $\mathbf{y}$ are modelled with help of a latent utility model u. Denote

$$u_r = \tilde{\eta}_r + \varepsilon_r$$

as the random utility function of the rth category, where $\tilde{\eta}_r$ is the utility of category r and ε_r is an error variable with cumulative distribution function C. The connection between $\mathbf{y}$ and u_r is

$$\mathbf{y} = r \quad \Leftrightarrow \quad u_r = \max_{s=1,\ldots,d+1} u_s.$$

Hence we observe the category r, when the associated utility is maximal. Using the extreme maximal-value distribution $C(x) = \exp(-\exp(-x))$ as the cumulative distribution of ε_r, we obtain (Fahrmeir et al., 2013)

$$P(y_i = r) = \frac{\exp \tilde{\eta}_{ir}}{\sum_{s=1}^{d+1} \exp \tilde{\eta}_{ir}}, \qquad r = 1, \ldots, d+1$$

Only the difference in utility functions is identifiable and therefore a reference category is required. With category $d+1$ as the reference category, we get

$$\pi_{ir} = \frac{\exp(\tilde{\eta}_{ir} - \tilde{\eta}_{i,d+1})}{1 + \sum_{s=1}^{d+1} \exp(\tilde{\eta}_{is} - \tilde{\eta}_{i,d+1})} = \frac{\exp \eta_{ir}}{1 + \sum_{s=1}^{d} \exp \eta_{is}},$$

where $\eta_{ir} = \mathbf{x}_i^T \boldsymbol{\beta}_r$ and $\boldsymbol{\beta}_r$ is the regression coefficients for category r. Recall, in logistic regression (d = 1), this probability is simplified as $\pi_i = \exp \eta_i / (1 + \exp \eta_i)$. With $d+1$ different categories, $\boldsymbol{\beta}$ is now a $k \times d$ coefficient matrix,

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_d \end{pmatrix}^T = \begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,d} \\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{k,1} & \beta_{k,2} & \cdots & \beta_{k,d} \end{pmatrix}.$$

For example, $\beta_{2,3}$ is the regression coefficient for the second variable in the third category. The linear predictor, η_{ir} could be seen as the log odds in logistic regression, as we model the ratio $P(y_i = 1)/P(y_i = 0)$. In nominal regression it can be seen as *relative risk* between the category r and the reference category $d+1$, as we model the ratio $P(y_i = r)/P(y_i = d+1)$ (Fahrmeir et al., 2013),

$$\eta_{ir} = \ln \frac{\pi_{ir}}{\pi_{i,d+1}}$$

The probabilities for an observation to belong to each category are as follows,

$$\pi_{ir} = \frac{\exp(\mathbf{x}_i^T \boldsymbol{\beta}_r)}{1 + \sum_{s=1}^d \exp(\mathbf{x}_i^T \boldsymbol{\beta}_s)}$$
$$\pi_{i,d+1} = 1 - \pi_{i1} - \cdots - \pi_{id} = \frac{1}{1 + \sum_{s=1}^d \exp(\mathbf{x}_i^T \boldsymbol{\beta}_s)}.$$

Keep in mind when interpreting β_{jr}, that a positive regression coefficient doesn't always mean an increase in the probability π_{ir} for an increasing x_{ij}. However, it can be seen as an increase in the odds relative to the reference category $d+1$. For example, let the coefficient for category s and variable j be greater than the coefficient for a category r and the same variable, i.e. $\beta_{js} > \beta_{jr}$. This tells us that the ratio $\pi_{is}/(\pi_{i,d+1})$ grows faster than the ratio $\pi_{ir}/(\pi_{i,d+1})$. Thus, there is a possibility for β_{jr} to have a decreasing effect on the probability π_{jr}, even though β_{jr} is positive (Fahrmeir et al., 2013).

2.4 Model Selection and Regularization

In this section we will describe some of the methods used to reduce the number of predictors in the model with minimal loss of information. This is called *model regularization*, or shrinkage. There are mainly two types of model regularization, *lasso regularization* and *ridge regularization*. The difference between these two models is in the form of what is called a penalty term.

We will also look at how to choose between different candidate models. Let k be the number of predictors we originally have and $P = \{x_1, x_2, \ldots, x_k\}$ be the set of predictors. Model selection is a way to identify which subset $P_0 \subset P$ that best explains the observed data. Common methods include using criteria like AIC, BIC, Mallows' C_p and adjusted R^2. The AIC is considered the first criterion to use in practice when it comes to deciding which model is better. In Section 2.4.2 we introduce AIC.

2.4.1 Lasso Regularization

A full model, i.e. a model using all p parameters, will almost always perform better than a model with less parameters on the particular data set. However, it will most likely overfit and lose accuracy when used on other data sets. Therefore we want to shrink the number of parameters as much as possible, without losing too much information. *Lasso regression* is very similar to the least squares method, but one additional penalizing term is added. Recall from Section 2.1.1 that the least squares method, which is defined as finding

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

was used to estimate the $\boldsymbol{\beta}$ in multiple linesar regression. The lasso regression is defined as finding

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{k} |\beta_j| \tag{2.11}$$

where as before, $\hat{y}_i = \beta_0 + \sum_{j=1}^{k} x_{ij}\beta_j$, and $\lambda \geq 0$ is a tuning parameter for the penalty term $\lambda \sum_{j=1}^{k} |\beta_j|$. The additional term forces the values of $\boldsymbol{\beta}$ towards zero. Since $\sum_{j=1}^{k} |\beta_j|$ is actually the L^1-norm of $\boldsymbol{\beta}$, we can also define the penalty term of lasso regression as $\lambda \sum_{j=1}^{k} |\beta_j| = \lambda \|\boldsymbol{\beta}\|_1$. Having $\lambda = 0$ results in the ordinary least squares method, and having $\lambda = \infty$ will force all of the regression coefficients to zero. Using an optimal value of λ will set coefficients for unnecessary variables to zero. The optimal value of λ can be found by cross-validation (see Section 2.1.4).

Now, why do we choose lasso regularization over ridge regularization? The ridge regression is defined as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{k} \beta_j^2$$

The last term is actually the squared L^2-norm, and can be redefined as $\lambda \sum_{j=1}^{k} \beta_j^2 = \lambda \|\boldsymbol{\beta}\|_2^2$. Because of this last term, ridge regression has a big disadvantage. The penalty $\lambda \|\boldsymbol{\beta}\|_2^2$ will shrink all of the coefficients towards zero (faster than lasso if the coefficients are far away from zero), but will not set any of them to excatly zero (unless $\lambda = \infty$) (James et al., 2013). This will result in having all the p coefficients in the final model. Since our goal is to shrink the number of parameters, we choose lasso regression where the regression coefficients can be set to zero. Reducing the number of parameters avoids overfitting, and by doing lasso regression this will hopefully be the case.

2.4.2 Akaike Information Criterion

Suppose we have several candidate models to choose between. How would we determine which model is better than the others? The intuition behind every statistical model, is to approximate reality. Given a data set, we would prefer the model that best approximates the data set, i.e. the model that has minimized the loss of information.

A measure of information loss was introduced by Kullback and Leibler (1951). If we knew the true distribution, $f(x)$, of the data, we could measure the information loss by Kullback–Leibler divergence. However, we do not know $f(x)$, and cannot know for certain which model suffers more loss than the other. Suppose we want to choose between a model $g_1(x)$ and $g_2(x)$. Akaike (1974) showed how much information is lost by $g_1(x)$ relative $g_2(x)$. The Akaike Information Criterion, or AIC, is an estimate of the amount of information lost by a given model, i.e. an estimate of the model quality. The AIC is defined by

$$\text{AIC} = -2k + 2\log L(\hat{\theta})$$

where k is the number of parameters included inn the model and $\hat{\theta}$ are a set of parameters that maximizes the likelihood L.

2.5 Tree-Based Ensemble Methods

Decision trees alone suffer from high variance. Tree-based ensemble methods are methods for reducing the variance. Say, we have observations $Y_1, \ldots, Y_n$ which are independent with variance σ^2. Then the mean of the observations, $\bar{Y}$, has variance σ^2/n. This means we can reduce the variance by averaging a set of independent observations. Tree-based ensemble methods are based on this idea, where we let an observation be a single tree.

2.5.1 Random Forests

First, let us introduce *bootstrap sampling*. We don't know the exact distribution of our data set, so have no distribution to sample from. What we do know, is that our data is coming from this unknown distribution. The solution is to use our data set as an empirical estimate of our distribution. Let n be the number of observations in the data set. Each of the observations are drawn with a probability of $\frac{1}{n}$, and the drawn observations together form the bootstrap sample. Random forests use bootstrapping to fit the trees (Breiman, 2001). The observations that don't make it to the training set are called "out-of-bag" observations, and are used for calculating the error. Each bootstrap sample is used to fit a single tree.

The idea is to combine the trees into a final model by taking the average. This is what we call *bagging*. If a predictor is clearly more significant than the others, it will most likely be included in the first split criterion for most of the fitted trees causing the trees to be correlated. This will result in a final model with high variance, which is not what we want. Random forests avoids this problem and we will see how later.

Now, why does the variance of the final model depend on the correlation between the trees? Consider we have the covariance between two trees ϕ_i and ϕ_j, $\text{Cov}(\phi_i, \phi_j) = \rho\sigma^2$ for $i \neq j$, where ρ is a positive correlation value and σ^2

is some constant variance. Denote M as the number of trees fitted. Then, the variance of the average is

$$\begin{aligned}
\text{Var}(\bar{\phi}) &= \text{Var}\left(\frac{1}{M}\sum_{b=1}^{M}\phi_i\right) \\
&= \sum_{b=1}^{M}\left(\frac{1}{M}\right)^2 \text{Var}(\phi_i) + 2\sum_{b=2}^{M}\sum_{j=1}^{b-1}\left(\frac{1}{M}\right)^2 \text{Cov}(\phi_i, \phi_j) \\
&= \left(\frac{1}{M}\right)^2 \sum_{b=1}^{M}\sigma^2 + 2\left(\frac{1}{M}\right)^2 \sum_{b=2}^{M}\sum_{j=1}^{b-1}\rho\sigma^2 \\
&= \frac{1}{M}\sigma^2 + 2\frac{M(M-1)}{2}\left(\frac{1}{M}\right)^2 \rho\sigma^2 \\
&= \frac{1}{M}\sigma^2 + \frac{(M-1)}{M}\rho\sigma^2 \\
&= \frac{1+(M-1)\rho}{M}\sigma^2
\end{aligned}$$

If we have highly correlated trees, i.e., ρ is close to 1, we can see that $\text{Var}(\bar{X})$ becomes close to σ^2. The variance will almost be equally high as for just one tree. If we have highly uncorrelated trees, i.e., ρ is close to 0, we see that $\text{Var}(\bar{X})$ becomes close to σ^2/M. Hence, averaging the trees has the effect we were originally looking for.

Let us see how random forests reduce correlation between the trees. Let k be the number of predictors we originally have, and $P = \{x_1, x_2, \ldots, x_k\}$ be the set of predictors. Instead of choosing between all k predictors when making a split, as in bagging, we choose a predictor from a subset $P_0 \subset P$ with size $z < k$. Typically, $z = \frac{k}{3}$ for regression and $z = \sqrt{k}$ for classification (James et al., 2013). Having $z = k$ would result in bagging. The subset P_0 is made by randomly selecting z predictors from P. For example, if $k = 9$, we would only consider $z = 3$ predictors in the next split. For example, $P_0 = \{x_2, x_3, x_9\}$ could then be the subset we were considering in the next split. The probability of having the most significant predictor in the next splitting criterion is lower, making the trees more uncorrelated. Thus, we will have a reduction of the variance in the final model. The algorithm for random forests is described in Algorithm 1.

Algorithm 1 Random Forest

```
1: for $m \in 1 : M$ do
2:     Draw a bootstrap sample from the training set.
3:     Make a decision tree, $\hat{f}^m$, based on the sample and only consider $z$
       predictors in each split.
   return $\hat{f}_{\text{RF}}(x) \leftarrow \frac{1}{M}\sum_{m=1}^{M}\hat{f}^m(x)$
```

2.5.2 Extreme Gradient Boosting

Before we discuss the theory behind Extreme Gradient Boosting (XGBoost), we start by introducing boosting in general. Boosting is another tree-based ensemble method used to improve the prediction. It is similar to bagging and random forests (see Section 2.5.1), in the way that they create a strong, final model based on a combination of decision trees. This is actually the main idea behind boosting, the idea of "creating a strong learner, based on a set of weak learners". Unlike random forests, boosting doesn't require bootstrapping. Instead, the trees are grown sequentially based on residuals (or misclassifications) from the previous tree. Fitting a tree to the residuals will make the model better in areas where it did not perform well (James et al., 2013). This way, the model learns slowly, but surely. Focusing on the residuals, not only reduce the dimensionality, but can also reduce the execution time as well.

The predicted class of observation i can be defined as

$$\hat{y}_i = \sum_{m=0}^{M} f_m(\mathbf{x}_i) = \theta_0 + \sum_{m=1}^{M} \theta_m \phi_m(\mathbf{x}_i), \tag{2.12}$$

(Chen and Guestrin, 2016), where the model from iteration m, f_m, corresponds to an independent basis function ϕ_m with weight θ_m, M is the number of tree structures added and $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ik}]$ is a vector of observed data for observation i.

The basis functions will be independent decision trees in the case of tree boosting algorithms, such that ϕ_m can be defined as in (2.2)

$$\phi_m(\mathbf{x}_i) = \sum_{t=1}^{T_m} \tilde{w}_{tm} I(\mathbf{x}_i \in R_{tm}), \tag{2.13}$$

where T_m is the number of leaf nodes in ϕ_m, R_{tm} is the tth region in ϕ_m and $\tilde{w}_{tm}$ is the weight of region R_{tm}. By inserting (2.13) into (2.12), we get

$$\begin{aligned}\hat{y}_i &= \theta_0 + \sum_{m=1}^{M} \theta_m \sum_{t=1}^{T_m} \tilde{w}_{tm} I(\mathbf{x}_i \in R_{tm}) \\ &= \theta_0 + \sum_{m=1}^{M} \sum_{t=1}^{T_m} w_{tm} I(\mathbf{x}_i \in R_{tm}),\end{aligned}$$

as the prediction of tree boosting algorithms. Here, we have merged θ_m and $\tilde{w}_{tm}$ into w_{tm}, because they are both weights for the leaf nodes for tree structure ϕ_m.

The set of basis functions $\Phi = \{\phi_1, \phi_2, \ldots, \phi_M\}$ are in this case the *weak learners*. These functions are greedily added to the final model, minimizing the following regularized objective

$$\{\hat{\phi}_m\} = \underset{\{\phi_m\}}{\arg\min} \sum_{i=1}^{n} \mathcal{L}(y_i, \hat{f}^{(m-1)}(\mathbf{x}_i) + \phi_m(\mathbf{x}_i)) + \Omega(\phi_m(\mathbf{x}_i)). \tag{2.14}$$

Here, $\mathcal{L}$ is a loss function, y_i is the true value of observation i, $\hat{f}^{(m-1)}$ is the model from the previous iteration and Ω is a function penalizing the complexity of the model. The temporary model $\hat{f}^{(m-1)}$ is simply all the basis functions, along with their respective weights, made before iteration m. Hence $\hat{f}^{(0)}$ will only result in the initial model $\theta_0 = \arg\min_\theta \sum_{i=1}^n \mathcal{L}(y_i, \theta)$, and $\hat{f}^{(M)}$ will result in the final model.

The loss function determines the quality of a predicted value. We compute the loss by measuring the discrepancy between the true value, picked by nature, and the the predicted value, $\hat{y}_i^{(m)} = \hat{f}^{(m-1)}(\mathbf{x}_i) + \phi_m(\mathbf{x}_i)$. Given the latter equation, we can generalize the loss function in the first term of (2.14), $\mathcal{L}(y_i, \hat{y}_i)$. Some common loss functions are

- $\mathcal{L}(y_i, \hat{y}_i) = \alpha(y - \hat{y}_i)^2$, quadratic loss function (for regression)
- $\mathcal{L}(y_i, \hat{y}_i) = I(y_i \neq \hat{y}_i)$, 0–1 loss (for classification)
- $\mathcal{L}(y_i, \hat{y}_i) = \exp(-\alpha y_i \hat{y}_i)$, exponential loss function (for binary classification)

where $\alpha > 0$ is some constant. The lower the value returned from the loss function, the better.

The complexity penalty term is the is defined by $\Omega(\phi_m) = \gamma T_m + \frac{1}{2}\nu\|w_m\|_2^2$ (Chen and Guestrin, 2016). The first term is penalizing the number of leaf nodes T through a penalizing factor γ. The second regularization term helps to smooth the final learnt weights through a parameter ν, to avoid overfitting. We can now expand the equation by rewriting (2.14) as the second order approximation and by inserting the expanded version of $\Omega(\phi_m)$,

$$\begin{aligned}\{\hat{\phi}_m\} = &\underset{\{\hat{\phi}_m\}}{\arg\min} \sum_{i=1}^{n} \left[L\left(y_i, \hat{y}_i^{(m)}\right) + g_i(\mathbf{x}_i)\phi_m(\mathbf{x}_i) + \frac{1}{2} h_i(\mathbf{x}_i)(\phi_m(\mathbf{x}_i))^2 \right] \\ &+ \gamma T_m + \frac{1}{2}\nu\|w_m\|_2^2.\end{aligned}$$

where $w_m = [w_{1m}, w_{2m}, \ldots, w_{T_m m}]$ and

$$\begin{aligned}\hat{g}(\mathbf{x}_i) &= \partial_{\hat{f}^{(m-1)}(\mathbf{x}_i)} L\left(y_i, \hat{f}^{(m-1)}(\mathbf{x}_i)\right), \\ \hat{h}(\mathbf{x}_i) &= \partial^2_{\hat{f}^{(m-1)}(\mathbf{x}_i)} L\left(y_i, \hat{f}^{(m-1)}(\mathbf{x}_i)\right)\end{aligned}$$

are the first and second order gradient statistics (Chen and Guestrin, 2016). The XGBoost algorithm can be referred to as *Newton tree boosting.* Newton tree boosting implements the second order approximation instead of only first order approximation, which are done by earlier boosting algorithms, such as *gradient boosting machine* (Friedman, 2001). The algorithm for XGBoost can be found in Algorithm 2.

The split is found by an exact greedy split finding algorithm (Chen and Guestrin, 2016). This algorithm starts from a single leaf and adds branches in

Algorithm 2 XGBoost

```
    Input: Data set, $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$
           A loss function, $\mathcal{L}$
           The number of iterations, $M$
           The maximum number of terminal nodes, $T$
           The learning rate, $\eta$
           A penalizing factor of number of leaf nodes, $\gamma$
           A weight-smoothing parameter, $\nu$
1: Initialize $\theta_0 = \arg\min_\theta \sum_{i=1}^n \mathcal{L}(y_i, \theta)$.
2: for $m = 1, 2, \ldots, M$ do
3:     $\hat{g}(\mathbf{x}_i) = \left[\frac{\partial L\left(y_i, \hat{f}^{(m-1)}(\mathbf{x}_i)\right)}{\partial \hat{f}^{(m-1)}(\mathbf{x}_i)}\right]$
4:     $\hat{h}(\mathbf{x}_i) = \left[\frac{\partial^2 L\left(y_i, \hat{f}^{(m-1)}(\mathbf{x}_i)\right)}{\partial \hat{f}^{(m-1)}(\mathbf{x}_i)^2}\right]$
5:     Determine the structure $\{\hat{R}_{tm}\}_{t=1}^{T_m}$ by selecting splits which maximize
         $\Gamma_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} \hat{g}(\mathbf{x}_i)\right)^2}{\sum_{i \in I_L} \hat{h}(\mathbf{x}_i) + \nu} + \frac{\left(\sum_{i \in I_R} \hat{g}(\mathbf{x}_i)\right)^2}{\sum_{i \in I_R} \hat{h}(\mathbf{x}_i) + \nu} - \frac{\left(\sum_{i \in I} \hat{g}(\mathbf{x}_i)\right)^2}{\sum_{i \in I} \hat{h}(\mathbf{x}_i) + \nu}\right] - \gamma$
6:     Determine the leaf weights $\{\hat{w}_{tm}\}_{t=1}^{T_m}$ for the learnt structure by
         $\hat{w}_{tm} = -\frac{G_{tm}}{H_{tm}} = -\frac{\sum_{i \in I_{tm}} \hat{g}_m(\mathbf{x}_i)}{\sum_{i \in I_{tm}} \hat{h}_m(\mathbf{x}_i)}$
7:     $\hat{f}_m(\mathcal{X}) = \eta \sum_{t=1}^{T_m} \hat{w}_{tm} I(\mathcal{X} \in R_{tm})$
8:     $\hat{f}^{(m)}(\mathcal{X}) = \hat{f}^{(m-1)}(\mathcal{X}) + \hat{f}_m(\mathcal{X})$
    Output: $\hat{f}(\mathcal{X}) = \hat{f}^{(M)}(\mathcal{X}) = \sum_{m=0}^M \hat{f}_m(\mathcal{X})$
```

an iterative manner. Let $I = I_L \cup I_R$, where I_L are the observations that fall into the left branch, and I_R are the observations that fall into the right branch. The loss reduction is then measured by

$$\Gamma_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \nu} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \nu} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \nu}\right] - \gamma. \tag{2.15}$$

(Chen and Guestrin, 2016). This equation is used in Algorithm 2 in line 5.

We will not go into detail about XGBoost. Instead we wil sum up some of the features that the boosting algorithm provides. XGBoost offers *complexity penalization* in terms of the function Ω. Complexity penalization is used to prevent overfitting and is one of the core improvements XGBoost has from earlier versions of boosting algorithms, such as gradient boosting machine (Friedman, 2001). XGBoost also provides *row subsampling*, yielding substantial performance improvements (Chen and Guestrin, 2016). Row subsampling works almost like bootstrap sampling, which is used in both bagging and random forests. The difference is that row subsampling draw samples from the data set without replacement, while bootstrapping draw samples with replacement. A row subsampling fraction $0 < \delta_r \leq 1$ is defined such that the sample size equals to $\delta_r n$, where n is the number of rows in the original data set (Nielsen, 2016). Having δ_r set to 1 will result in a procedure with no row subsampling. Similarly, we have *column subsampling*. This is a simple, yet effective technique which is also borrowed from random forests (Chen and Guestrin, 2016). This helps to decorrelate the trees ϕ_m, which reduces the variance of the fitted model. A column subsampling fraction $0 < \delta_c \leq 1$ is defined such that the sample size equals to $\delta_c k$, where k is the number of columns in the data set. Having δ_c set to 1 will result in a procedure with no column subsampling. Compared to earlier versions of boosting, gradient boosting machine only includes row subsampling.

In addition to this, number of leaf nodes for each tree is not fixed, but optimized for the respective tree (Nielsen, 2016). The weight on each leaf in a tree structure ϕ_m is shrunk independently and not by the same factor. Leaf weights that are estimated on less data will be shrunk more than other leaf weights. We also have a *learning rate* $0 < \eta \leq 1$, also called shrinkage parameter, that is the step length at each iteration (Nielsen, 2016). It determines how much we want to learn from the trees that are added to the model. The lower this rate is, the slower the algorithm learns. A lower value of η tends to improve generalization performance, but requires a higher value of number of iterations M. XGBoost models can also be trained on data set with missing values (Chen and Guestrin, 2016).

2.6 ROC Curves and AUC

In tree-based ensemble methods there are no likelihoods and regression coefficients, hence we no longer have criteria like AIC, BIC and Mallows' C_p when it comes to assessment of such models. We need to evaluate the models in a different way, but keep in mind that we somehow want to compare these models

		Predicted class	
		N	**P**
Actual class	**N**	True Negative	False Positive
	P	False Negative	True Positive

Table 4: A visualization of a confusion matrix for binary classification. An observation whose actual class is true, is called a true positive (TP) if it is predicted true and a false negative (FN) if it is predicted false. Similarly, an observation whose actual class is negative, is called a false positive (FP) if it is predicted true and a true negative (TN) if it is predicted false. Thus, we want the number of true negatives and true positives to be as high as possible.

to regular GLM models. Thus, focusing on the predicted probabilities seems like a good alternative here.

One way to deal with is to use receiver operating characteristic curves, or *ROC curves*. ROC curves are based on the predicted probabilities and their actual classes. Before we explain the ROC curve, let us introduce the *confusion matrix*. This is a $K \times K$ matrix that keeps count of the number of successively and wrongly predicted classes, based on the predicted and actual class of each observation. The confusion matrix for binary classification, i.e. $K = 2$, is visualized in Table 4. The ROC curve measures two rates called *sensitivity* (true positive rate) and *specificity*, (true negative rate) defined as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.16}$$

where TP is the number of true positives, FN is the number of false negative, TN is the number of true negatives and FP is the number of false positives.

Let $\tau \in \{0, 1\}$ be a parameter and $\hat{y}_i$ the probability that observation i belongs to class 1. An observation belongs to class 1 if $\hat{y}_i \geq \tau$. ROC curves are made by computing the sensitivity and the specificity for every value of τ. This results in a parametric curve with τ as the parameter, with specificity on the x-axis and sensitivity on the y-axis. An example with 20 observations, where $\hat{y}_i$ are the probability for observation i to be classified as 1 and $y_i \in \{0, 1\}$ are the actual class, are shown in Table 5. The corresponding ROC curve is visualized in Figure 5.

Starting with $\tau = 0$, every $\hat{y}_i$ will be classified to class 1. Since no observation is classified as 0, the number of false negatives and true negatives will be zero,

i	1	2	3	4	5	6	7	8	9	10
$\hat{y}_i$	0.92	0.30	0.51	0.71	0.86	0.97	0.62	0.71	0.13	0.97
y_i	1	1	0	1	1	1	1	0	1	1
i	11	12	13	14	15	16	17	18	19	20
$\hat{y}_i$	0.52	0.98	0.81	0.52	0.45	0.24	0.11	0.24	0.43	0.45
y_i	0	1	1	1	0	0	0	0	1	0

Table 5: An example of vectors that can be used to make a ROC curve. In this case, we have 20 observations. $\hat{y}_i$ is the probability that observation i belongs to class 1, and y_i is the true class of observation i.

i.e. FN = TN = 0. This results in a sensitivity equal to one, and a specificity equal to zero, according to the respective formulas in (2.16). Thus, we start in the top right corner of the ROC plot in Figure 5. As we move τ towards one, each observation $\hat{y}_i$ at a time will consequently be classified as 0. The sensitivity and specificity will consequently move towards zero and one, respectively.

From the example in Table 5 and Figure 5, the red dot in the top right corner corresponds to $0 \leq \tau < 0.11$. We refer to this red dot as a_1. When $\tau = 0.11$, the observation $\hat{y}_{17}$ are classified as 0. Since this observation has 0 as the actual class, the number of true negatives becomes equal to one, i.e. TN = 1. The specificity is then increased from 0/8 to 1/8 = 0.917 and the ROC curve moves to the left. The sensitivity stays the same, because it does not depend on true negatives. We refer to the red dot to the left of a_1 as a_2. Point a_2 corresponds to $0.11 \leq \tau < 0.13$. When $\tau = 0.13$, the observation $\hat{y}_9$ are also classified as 0. Since this observation has 1 as the actual class, the number of false negatives becomes equal to one, i.e. FN = 1. The sensitivity is then decreased from 12/12 to 11/12 = 0.917 and the ROC curve moves down. The specificity stays the same, because it does not depend on false negatives. The ROC plot in Figure 5 is the resulting sensitivity and specificity when gliding the parameter τ through the interval $[0, 1]$.

The closer this curve is to the top left corner, the better is the model. Similarly, the more area that is covered under the curve, the better is the model. This measure is called *area under curve*, or AUC, and will take a value between 0.5 and 1. Having an AUC equal to one will be a perfect classification. Hence, we usually prefer the model that has AUC close to one, but in applications more weight may be given to sensitivity or to specificity, or one or both might be constrained.

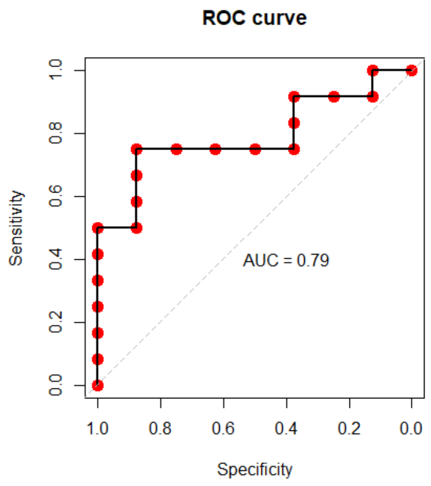


Figure 5: A visualization of a ROC curve which is made from the example data in Table 5. The y-axis shows sensitivity and x-axis specificity. For each value of $\tau \in [0, 1]$, the specificity and sensitivity are calculated. The red dots correspond to each observation. The corresponding AUC is 0.79.

3 Implementation in R

Here we will describe how we use the theory in Section 2 to attack the problems given in the introduction. First of all, we describe the data sets `old` and `new`, and their features, in Section 3.1. We will then explain how we clean the data set and perform nominal regression in R, as well as how to do the analysis. This is done in Section 3.2. In Section 3.3 we start working on the bigger task, i.e. the indicator model. Here, we again start by cleaning the data set, before we perform logistic regression. The regression coefficients will be useful for interpretation of the regression model. We then perform different methods for variable selection in order to reduce the dimensionality and execution time, and at the same time increase the performance of the model. At last, we explain how a ROC curve can be drawn in R, as well as calculating the AUC for a specific model.

3.1 Understanding the Data Sets

The data sets are provided by the bank itself. Each row of the data set represents a customer of the bank. Each customer is described by $k = 45$ variables. Table 6 shows a list of the variables. We have two data sets with these variables, extracted from different dates. One of them is from 31 January 2018, and will be referred to as `old`. The other data set is from 31 January 2019, and will be referred to as `new`. The number of customers in the data set `old` is $n_{\text{old}} = 69371$. The number of customers in the data set `new` is $n_{\text{new}} = 78940$. The most important column, which will play a huge role in the analysis, is the variable number 42, `Oppsummering.klasse`. This column shows the customer category for the different customers. The customer categorization is described in Table 2.

Table 6: A list of all the variables in the data set `old`.

Nr	Name	Description
1	`KUNDE_NR_HASH`	An unique customer identification key.
2	`status`	A status of a customer consisting of 4 degrees of activity. • Active last five years • Active last year • Passive last five years • Without active account
3	`innskudd`	A binary variable denoting whether the customer has positive balance on deposit account (1) or not (0).

Continued on next page

Table 6 – *Continued from previous page*

Nr	Name	Description
4	`utlan`	A binary variable denoting whether the customer has loan (1) or not (0).
5	`medlantaker`	A binary variable denoting whether the customer is a loan co–signer (1) or not (0).
6	`SLUTT_DATO`	The date when information of the customer was extracted.
7	`BK_ANSVARSTED_KODE`	A 6 digit code which is different for employees, members and non-members. Says something about the customer affiliation.
8	`Postkode`	The ZIP code of the customer.
9	`BK_KJONN_KODE`	The sex of the customer. The customer is either male (M) or female (K).
10	`ALDER`	The age of the customer.
11	`KUNDE_START_DATO`	The date when a customer became a customer of the bank.
12	`KUNDE_SLUTT_DATO`	The date when a customer ended the relationship with the bank.
13	`DOD`	A binary variable denoting whether the customer has died (1) or not (0).
14	`LAN`	A counting variable denoting the number of loaning products.
15	`INNSKUDD`	A counting variable denoting the number of deposit accounts.
16	`DEBETKORT`	A counting variable denoting the number of debit cards.
17	`KREDITTKORT`	A counting variable denoting the number of credit cards.
18	`NETTBANK`	A binary variable denoting whether the customer has activated internet banking (1) or not (0).

Continued on next page

Table 6 – *Continued from previous page*

Nr	Name	Description
19	`MOBILBANK`	A binary variable denoting whether the customer has activated mobile banking (1) or not (0).
20	`BANKID`	A binary variable denoting whether the customer has activated BankID (1) or not (0).
21	`SALDO.INNSKUDD`	The balance on the deposit account of the customer.
22	`Medlem/ikke.medlem`	A binary variable denoting whether the customer is a member (1) or not (0).
23	`Lønnsinngang`	A binary variable denoting whether the customer has payroll input (1) or not (0).
24	`Søkeprosess`	A binary variable donting whether the customer has applied for loan/ credit card (1) or not (0).
25	`Transaksjoner.siste.år`	The number of transactions the last year.
26	`Transaksjoner.nest.siste.år`	The number of transactions the second last year.
27	`Transaksjoner.tredje.siste.år`	The number of transactions the third last year.
28	`Transaksjoner.fjerde.siste.år`	The number of transactions the fourth last year.
29	`Transaksjoner.femte.siste.år`	The number of transactions the fifth last year.
30	`Input.aktivitet.kredittkort`	A binary variable denoting whether the customer has one or more active credit cards. A credit card is active if it has been used within the last six months or if it has a limit of over NOK 1000.
31	`Fylke`	The county of the customer. This variable has 18 levels.

Continued on next page

Table 6 – *Continued from previous page*

Nr	Name	Description
32	`Tot_medlem`	This variable says something about whether the customer is a member of the bank (1) or not (0). Together with variable `Medlem/ikke.medlem`, it is used to form the variable `Medlem`.
33	`Aktivmedlem`	A member whose company has pension scheme at the bank.
34	`Skade_avtale`	A binary variable denoting whether the customer has damage insurance (1) or not (0).
35	`Bank_avtale`	A binary variable denoting whether the customer is a customer of the bank (1) or not (0). From this definition, every customer should be denoted 1, as everyone in the data set is a customer of the bank. However, customers under the age of 18 is not set to 1 (and due to some delay in the system).
36	`Nyhetsbrev`	A binary variable denoting whether the customer receives newsletters.
37	`KRA_Livsfase`	The phase of life of the customer. Customers are divided into 10 different levels: • Single parent • Established family with children • Couple without children • Middle aged single • Middle aged couple • Senior single • Senior couple • Family with toddlers • Single • Student

Continued on next page

Table 6 – *Continued from previous page*

Nr	Name	Description
38	`KRA_Bolig_By_land`	The customer lives in either an area where the people live far from each other (S), close to each other (T) or in the city (B).
39	`KRA_Boligtype_Enkel`	The type of dwelling. Customers are divided into 8 different levels: • Detached • Semi-detached • Buildings with shared accommodations • Holiday cottage • House with garage and/or other outbuildings. • Commercial buildings and other buildings • Terraced house, linked house and other small houses. • Apartment in high-rise buildings .
40	`BIL_I_HUSTAND`	A binary variable denoting whether the customer has a car registered on the household (1) or not (0). For example, someone in the family could own a car, and it would still be denoted as 1.
41	`Fond_avtale`	A binary variable denoting whether the customer has a fund (1) or not (0).
42	`Oppsummering.klasse`	The customer categorization as defined in Table 2. Customers are divided into six classes, A–F.
43	`Medlem`	The customer is either a member (Medlem), non-member (Ikke-medlem) or non-member with member benefits (Ikke-medlem med medlemsbetingelser).

Continued on next page

Table 6 – *Continued from previous page*

Nr	Name	Description
44	`AvtaleGiro`	A binary variable denoting whether the customer has activated direct debit (1) or not (0).
45	`eFaktura`	A binary variable denoting whether the customer has activated eFaktura (electronical billing) (1) or not (0).
47	`Tlfsamtaler.siste.år`	Number of calls recieved from the customer within the last year.

Some of the variables listed above, are variables counting the number of transactions for the previous years. Transactions can be seen as some sort of the activity of the customer, and the more transactions that are registered, the more active is the customer. Having more transactions one year, and less transactions the second year, can be seen as a decrease in activity. A decrease in activity could mean that this customer will end up in a more inactive category than the current category in the future. We create a new variable, `Transer.diff.0.1`, which will be defined as the difference

$$\texttt{Transaksjoner.siste.år} - \texttt{Transaksjoner.nest.siste.år},$$

i.e., the difference between variable number 25 and 26 in Table 6. This allows us not only to see if they have decreased or increased in transactions, but also by what amount. A negative value will be a decrease in transactions, while a positive value will be a increase in transactions. We will add this to our list of variables that we will use. This variable is a linear combination of two other variables, and having all of these three variables will result in an infinite number of solutions of the least squares method. Thus, we need to remove one of these three variables, which will be the variable `Transaksjoner.nest.siste.år`.

Also, by a proposal from the bank employees, we choose to look at interaction terms between the variables `innskudd`, `utlan` and `Input.aktivitet.kredittkort`. Thus, we get four new variables to take care of,

- `utlan0:Input.aktivitet.kredittkort1`
- `innskudd0:utlan0`
- `innskudd0:Input.aktivitet.kredittkort1`
- `innskudd0:utlan0:Input.aktivitet.kredittkort1`

We already have the variables

- `innskudd1:utlan1:Input.aktivitet.kredittkort0` (within intercept)
- `innskudd0`
- `utlan0`
- `Input.aktivitet.kredittkort1`

thus we have all 8 combinations of `Input.aktivitet.kredittkort`, `innskudd` and `utlan`.

3.2 Significant Differences Between Customer Categories

For this task we are going to use nominal regression, which we introduced in Section 2.3, (see Section 1 for task descriptions). Nominal regression models use unordered categories as the response. The categories A–F, which is found in the variable `Oppsummering.klasse`, are somewhat ordered categories, but for simplicity we will assume they are unordered. We want to know if there are any significant differences between these categories, and we will use the output regression coefficients from nominal regression to solve this task. The analysis will be based on the `new` data set.

3.2.1 Column Filtering

Before applying any methods to the data set, we start off by removing unnecessary variables. These variables can be identical or unique for each of the n customers, i.e. columns containing only one or n different values, respectively. They could also be linearly dependent of other variables or "better" explained by other variables. Linearly dependent variables will result in infinite number of solutions of the least squares method. Thus, there will be no unique solution for the coefficients estimates $\hat{\boldsymbol{\beta}}$ for such a model.

The response variable `Oppsummering.klasse` are mainly calculated based on formulas including other variables in the data set. The criteria are shown in Table 2. Hence, before we start analyzing, we need to remove these variables also. These variables will turn out to be very significant if we do not remove them and they will most likely make us miss out on other potentially significant variables. The variables that are included in the category formulas are `status`, `innskudd`, `utlan`, `medlantaker`, `DEBETKORT`, `NETTBANK`, `MOBILBANK`, `BANKID`, `KREDITTKORT`, `SALDO.INNSKUDD` and `Lønnsinngang`. Thus, these variables are being removed from the data set `new`. In Table 7 we show which variables that is removed from the data set, and explain the reason why they are being removed.

Table 7: Variables to be removed from the data set `task1`.

Nr	Name	Reasons to remove variable
1	`KUNDE_NR_HASH`	This is a unique text string for every customer, and makes no sense to bring in to an analysis.
2	`status`	Partly explained by the variable `Oppsummering.klasse`.
3	`innskudd`	Partly explained by the variable `Oppsummering.klasse`.
4	`utlan`	Partly explained by the variable `Oppsummering.klasse`.
5	`medlantaker`	Partly explained by the variable `Oppsummering.klasse`.
6	`SLUTT_DATO`	The data of every customer was extracted the same day, i.e. 31/01/2018 for the dataset `old` and 31/01/2019 for the dataset `new`.
7	`BK_ANSVARSTED_KODE`	This variable is partly explained by the variable `Medlem`.
8	`Postkode`	There are 3029 different ZIP codes in the dataset. We also have another area variable, `Fylke`, which is only 18 different codes. Hence we choose to remove this variable, and keep `Fylke`.
12	`KUNDE_SLUTT_DATO`	No customer in the dataset has ended the relationship with the bank. Hence, all of the customers denoted as 0.
13	`DOD`	All of the customers denoted as 0.
14	`LAN`	Partly explained by the variable `utlan`.
15	`INNSKUDD`	Partly explained by the variable `innskudd`.

Continued on next page

Table 7 – *Continued from previous page*

Nr	Name	Reasons to remove variable
16	`DEBETKORT`	A product provided by the bank. Therefore, partly explained by the variable `Oppsummering.klasse`.
17	`KREDITTKORT`	A product provided by the bank. Therefore, partly explained by the variable `Oppsummering.klasse`.
18	`NETTBANK`	A product provided by the bank. Therefore, partly explained by the variable `Oppsummering.klasse`.
19	`MOBILBANK`	A product provided by the bank. Therefore, partly explained by the variable `Oppsummering.klasse`.
20	`BANKID`	A product provided by the bank. Therefore, partly explained by the variable `Oppsummering.klasse`.
21	`SALDO.INNSKUDD`	Partly explained by the variable `Oppsummering.klasse`.
22	`Medlem/ikke.medlem`	Used, together with `Tot_medlem`, to create the variable `Medlem`.
23	`Lønnsinngang`	Partly explained by the `Oppsummering.klasse`.
25	`Transaksjoner.siste.år`	Number of capital transactions. Therefore, partly explained by the variable `Oppsummering.klasse`.
26	`Transaksjoner.nest.siste.år`	We only consider the number of transactions within the last year.
27	`Transaksjoner.tredje.siste.år`	We only consider the number of transactions within the last year.
28	`Transaksjoner.fjerde.siste.år`	We only consider the number of transactions within the last year.

Continued on next page

Table 7 – *Continued from previous page*

Nr	Name	Reasons to remove variable
29	`Transaksjoner.femte.siste.år`	We only consider the number of transactions within the last year.
30	`Input.aktivitet.kredittkort`	Partly explained by the variable `Input.aktivitet.kredittkort`.
32	`Tot_medlem`	Used, together with `Medlem/ikke.medlem`, to create the variable `Medlem`.
33	`Aktivmedlem`	Used to create `Tot_medlem`, which again is used to create `Medlem`.
35	`Bank_avtale`	Every customer in the datasets is a customer of the bank, hence all of the customers should be denoted as 1.

3.2.2 Row Filtering

The next thing that needs to be done, is to remove all rows with missing values. If the data is missing at random, then it is safe to remove the corresponding rows. However, if the missing values are not random, then removing the row can produce a bias. Since our data most likely are missing at random (through personal communication with lead business analyst of the bank), we can go ahead and remove the corresponding rows, which can simply be done by using the R function `complete.cases()`. This leaves us a simplified smaller data set, which we will call `task1`. This data set has $n_{\text{task1}} = 45384$ bank customers that is explained by $k_{\text{task1}} = 16$ variables.

3.2.3 Dummy Variable Coding

Categorical variables are transformed into numerical columns, by using a technique called *dummy variable coding*. Denote by κ_j the number of categories within a categorical variable j. This technique transform one column with κ_j categorical classes, to κ_j binary columns. These columns are given a high (1) and a low (0) value based on if the specific binary column matches the category in the old categorized variable. Let us look at an example. Say, we have a variable `color`, whose categories are `blue`, `red` and `yellow`. We will refer to these categories as *levels*. Here, the number of levels is $\kappa = 3$. Thus, we transform the category variable `color` into three new binary columns, one for each level. An illustration is shown in Table 8.

Due to many variables being categorical in the `task1` data set, some with more levels than others, we end up with more than $k = 16$ regression coeffi-

Dummy variable coding

color		color:blue	color:red	color:yellow
blue		1	0	0
red		0	1	0
yellow		0	0	1
yellow	$\Longrightarrow$	0	0	1
red		0	1	0
red		0	1	0
blue		1	0	0
red		0	1	0

Table 8: This is an illustration of dummy variable coding. A variable with categories are transformed into columns with binary values corresponding to each category.

cients. For example, the variable `Fylke` has $\kappa_{31} = 18$ levels and the variable `KRA_Livsfase` has $\kappa_{37} = 10$ levels. All in all, transforming into dummy variable coding results in a total of $k_{\text{task1}} = 48$ individual variables.

3.2.4 Nominal Logistic Regression in R

Multiple different packages are available in order to perform nominal regression. Some of them are the `mlogit()` function from the `mlogit` package, the `multinom()` function from the `nnet` package and the `vglm()` function from the `VGAM` package (Croissant, 2019; Ripley and Venables, 2016; Yee, 2019). Here, we will use the latter alternative. The `vglm()` function can perform both nominal and ordinal regression, by setting the argument `family = multinomial` and `family = cumulative`, respectively. To obtain the regression coefficients, $\boldsymbol{\beta}$, one can simply use the `summary()` function on the vglm-object.

We have six customer categories as the response, which results into $d = 5$ in the nominal model. Category 1 in the output corresponds to customer category F, category 2 in the output corresponds to customer category E and so on. Category $d + 1$ corresponds to customer category A, which is our reference category. Thus, 48×5 coefficients are being output when performing nominal regression.

3.3 Indicator Model

For this task, (see Section 1 for task descriptions), we are going to develop an indicator that indicate whether an active customer (category A, B, C, D) soon will be inactive (category E, F) or not. This is the task that we emphasized the most. To attack this problem, we first need to make the response variable.

The response variable will be 1 if the customer has gone from an active category to an inactive category, and 0 otherwise. Also, if a customer from the `old` data set are not in the `new` data set, means that the customer is not a customer anymore. Hence, we will also denote the customer as 1 if the customer has an active category in the `old` data set, and is not found in the `new` data set. The response variable can easily be created by using the variable `Oppsummering.klasse` from both data sets.

This variable will be called `Response`, and will be merged with the `old` data set. We can now apply methods, such as variable selection through AIC and lasso regularization, to the data set.

3.3.1 Column Filtering

Again, before deriving any methods on the data set, we start off by removing unnecessary variables. These variables can be identical or unique for each of the n customers, i.e. columns containing only one or n different values, respectively. They could also be linearly dependent of other variables or "better" explained by other variables. Linearly dependent variables will result in infinite number of solutions of the least square method. Thus, there will be no unique solution of $\hat{\boldsymbol{\beta}}$ for such a model.

In Section 3.2.1 we removed the variables that explained the `Oppsummering.klasse` variable. The bank customers are better explained with many specific variables, than one "universal" variable. In the first task, the `Oppsummering.klasse` variable was required, since this was the response variable. In this task, this variable is just a variable like any other. Hence, we will do the opposite in this task, i.e. remove the the `Oppsummering.klasse` variable and keep the variables `status`, `innskudd`, `utlan`, `medlantaker`, `DEBETKORT`, `NETTBANK`, `MOBILBANK`, `BANKID`, `Transaksjoner.siste.år`, `KREDITTKORT`, `SALDO.INNSKUDD` and `Lønnsinngang`. In Table 9 we show which variables that is removed from the data set, and explain the reason why they are being removed.

Table 9: Variables to be removed instantly from the data set `task2`

Nr	Name	Reasons to remove variable
1	`KUNDE_NR_HASH`	This is a unique text string for every customer, and makes no sense to bring in to an analysis.
6	`SLUTT_DATO`	The data of every customer was extracted the same day, i.e. 31/01/2018 for the data set `old` and 31/01/2019 for the data set `new`.
7	`BK_ANSVARSTED_KODE`	This variable is partly explained by the variable `Medlem`.

Continued on next page

Table 9 – *Continued from previous page*

Nr	**Name**	**Reasons to remove variable**
8	`Postkode`	There are 3029 different ZIP codes in the dataset. We also have another area variable, `Fylke`, which is only 18 different codes. Hence, we choose to remove this variable and keep `Fylke`.
12	`KUNDE_SLUTT_DATO`	No customer in the dataset has ended the relationship with the bank. Hence, all of the customers denoted as 0.
13	`DOD`	All of the customers denoted as 0.
22	`Medlem/ikke.medlem`	Used, together with `Tot_medlem`, to create the variable `Medlem`.
26	`Transaksjoner.nest.siste.år`	A linear combination of variable `Transaksjoner.siste.år` and `Transer.diff.0.1`.
27	`Transaksjoner.tredje.siste.år`	We only consider the number of transactions within the last year.
28	`Transaksjoner.fjerde.siste.år`	We only consider the number of transactions within the last year.
29	`Transaksjoner.femte.siste.år`	We only consider the number of transactions within the last year.
32	`Tot_medlem`	Used, together with `Medlem/ikke.medlem`, to create the variable `Medlem`.
33	`Aktivmedlem`	Used to create `Tot_medlem`, which again is used to create `Medlem`.
35	`Bank_avtale`	Every customer in the data set is a customer of the bank, hence all of the customers should be denoted as 1.

Continued on next page

Table 9 – *Continued from previous page*

Nr	Name	Reasons to remove variable
42	`Oppsummering.klasse`	This variable is based on the variables `status`, `innskudd`, `utlan`, `medlantaker`, `Transaksjoner.siste.år`, `KREDITTKORT`, `SALDO.INNSKUDD` and `Lønnsinngang`, as explained in Table 2.

3.3.2 Row Filtering

To simplify our data set, we can remove all inactive customers, i.e. customers within categories E and F, from the `old` data set. These customers already are inactive in the `old` data set, which makes it impossible for them to go from being active to be inactive. This will also help to reduce nuisance, as these customers are irrelevent for the model. Furthermore, we go ahead and remove rows with missing values, which can simply be done by using the `R` function `complete.cases()`. This leaves us a simplified smaller data set, which we will call `task2`. This data set has $n_{\text{task2}} = 26189$ bank customers that is explained by $k_{\text{task2}} = 33$ variables.

3.3.3 The full model

To get a quick overview of the model containing all the $k_{\text{task2}} = 33$ variables, i.e. the *full model*, we can simply run the `glm()` function on the data set. Our response is binary, hence a logistic regression model is preferred. This can be done by including the argument `family = "binomial"` in the `glm()` function. The function `summary()` can be used on the fitted object in order to obtain the regression coefficient estimates, along with their standard error and p-value. We can then easily sort out the the most significant coefficients with the lowest p-values. Dummy variable coding is also used to transform category variables into numeric variables, which results in $k_{\text{task2}} = 71$ variables for the `task2` data set. See Section 3.2.3 for the explanation of dummy variable coding. The results are shown in Section 4. The next two sections are using this glm-model as a base model to perform variable selection.

3.3.4 Variable Selection Based on AIC

There are multiple ways to select the best subset of variables. There are mostly greedy, stepwise algorithms that use criteria like AIC, BIC, Mallows' C_p and adjusted R^2. Several functions in `R` can be used for model selection, which is the `step()` function (or the `stepAIC()` function from the `MASS` package), the `regsubsets()` function from the `leaps` package and `RegBest()` from the

FactoMineR package. We will use the stepwise algorithm called stepAIC() function. This function takes in and examines a range of models, and calculates the AIC for these models in an iterative manner (Ripley et al., 2013). We choose to examine the full range, i.e. from including all the variables, to only examine the intercept. There are three different directions to choose, “forward”, “backward” and “both”, where “both” is the default for the stepAIC() function.

A forward stepwise model selection means that we start with no variables in the model, then calculate the AIC on the k different models, by iteratively adding one variable to the model at a time. The variable that resulted in the lowest AIC are being added to the model. The next iteration we examine $k-1$ different models and add the variable that result in the lowest AIC, and so on. A backward stepwise model selection means that we start with all of the variables, then calculate the AIC on the k different models, by iteratively removing one variable from the full model at a time. The variable that results in the highest AIC are being removed from the model. The next iteration we examine $k-1$ different models, and then remove the variable that resulted in the highest AIC. The forward and backward stepAIC algorithm are described in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 Forward stepAIC

```
1: minAIC = $\infty$
2: for $i \in k : 1$ do
3:     minAIC2 = $\infty$
4:     for $j \in 1 : i$ do
5:         Add the $j$th variable (that is not in the model) and calculate the AIC
           for this model, $f_{ij}(x)$.
                           AIC = AIC($f_{ij}(x)$).
6:         minAIC2 = min(AIC, minAIC2)
7:         minAIC = min(AIC, minAIC)
8:     Add the variable that resulted in minAIC2 to the model.
   return Model $f_{ij}(x)$ that resulted in minAIC.
```

We choose to do both directions, where the output will be the model with the lowest AIC from the two algorithms. Running the stepAIC() on the model from Section 3.3.3 will return the suggested model based on AIC.

3.3.5 Variable Selection Based on Lasso Regularization

We can simply use the glmnet() function from the glmnet package to perform lasso regression. The coefficients that are forced to be exactly zero, will be excluded from the suggested model. The glmnet() function has an argument, alpha, that determines what type of regression we are going to use (James et al., 2013). Ridge regression is fit when alpha = 0 and lasso regression is fit when alpha = 1, thus we set alpha = 1. This function also takes in the

Algorithm 4 Backward stepAIC

```
1: minAIC = $\infty$
2: for $i \in k : 1$ do
3:     maxAIC = $-\infty$
4:     for $j \in 1 : i$ do
5:         Remove the $j$th variable (that is in the model) and calculate the AIC
           for this model, $f_{ij}(x)$.
               $$\text{AIC} = \text{AIC}(f_{ij}(x)).$$
6:         maxAIC = max(AIC, maxAIC)
7:         minAIC = min(AIC, minAIC)
8:     Remove the variable that resulted in maxAIC from the model.
   return Model $f_{ij}(x)$ that resulted in minAIC.
```

penalty value λ as an argument. The optimal value for λ can be found by cross validation by using another function within the `glmnet` package, which is the function `cv.glmnet()`.

There are two different values of λ that are returned from the `cv.glmnet()` function, namely $\lambda_{1\text{se}}$ and λ_{min}. The latter is defined as the λ that minimizes the mean cross-validated error. The $\lambda_{1\text{se}}$ is the largest value of λ such that the error is within one standard error of the minimum (Hastie and Qian, 2014). Now, which of the λ's should we use? Based on the cross-validation, having $\lambda = \lambda_{\text{min}}$ results in the best model. However, this model tends to be more complex, i.e. having more variables, and probably overfit. Having $\lambda = \lambda_{1\text{se}}$ we would get a simpler model including less variables, but then again it would perform slightly worse than the model resulting from λ_{min}. In our case, performing lasso regression with $\lambda = \lambda_{\text{min}}$ results in only three of $k_{\text{task2}} = 71$ coefficient being set to zero. Setting $\lambda = \lambda_{1\text{se}}$ results in getting 47 regression coefficients set to zero. Getting a much simpler model at the cost of some cross-validation error is preferred, since we want to observe which variables are less significant than other. Hence, we choose to set $\lambda = \lambda_{1\text{se}}$. The results can be found in Section 4.2.2.

3.3.6 Variable Selection Based on Variable Importance

Variable importance is another way to determine which variables to use in a model. This is a feature from tree-based ensemble methods, which allows us to see the importance of each variable. In the case of using regression trees, the importance are measured in decrease of RSS averaged over all M trees, and for classification trees, the importance are measured in decrease of the Gini index averaged over all M trees (James et al., 2013).

We simply use the `randomForest()` from the `randomForest` package to perform random forests. Along with the predicted classes (or probabilities for being classified to class 1), we can plot the variable importance by using the `varImpPlot()` function, which is also a part of the `randomForest` package, on the random forest object.

3.3.7 Prediction Methods and Model Assessment

We have seen how to select variables. The next thing to do is to select a statistical method. An important key to constructing an accurate model is to decide, for any given data set X, which method produces the best results (James et al., 2013). We will use different methods, such as binary logistic regression, random forests and XGBoost, to construct our models. For these methods, we are going to use the `glm()` function, `randomForest()` function from the `randomForest` package and `xgb.train()` function from the `xgboost` package, respectively, to train our models (Liaw and Wiener, 2019; Chen et al., 2019). We could also fit a lasso model as a fourth method. Lasso models are based on regression, which lacks of prediction accuracy compared to tree-based ensemble methods Hence, we choose to only include binary logistic regression as the only regression method.

The predicted values can be found through the `predict()` function, by inserting the model object and test data. The output are the probabilities for each customer to be inactive in the future. Finally, we will evaluate the outcome by measuring the AUC value based on ROC-plots, (see Section 2.6). This can simply be done by using the `roc()` function from the `pROC` package (Robin et al., 2019). This will return a ROC-object. We can plot the ROC-object by directly using the `plot()` function on the object, and get the AUC value by using the `auc()` function, from the `pROC` package, on the same object (Robin et al., 2019). We will decide the best model based on the AUC values and the ROC plots.

To avoid lucky and unlucky results, i.e. very good or very bad performance for a particular partition of the training and test set, when deciding the best model, an approach could be to compute the average test AUC over N_{sim} simulations. This way, we can observe which models performance is more robust than others. The model that perform best on average, will be strongly recommended as the indicator model.

4 Results and Discussion

In this section we will show and discuss the results from Section 3. The results from the smaller task are shown in Section 3.2, and the results from the bigger task are shown in Section 3.3.

4.1 Significant Differences Between Groups

In Section 3.2.4 we performed nominal regression in R, with the customer categories A–F as the response. The variables that show the most significance between categories are listed in Table 10. The full output can be found in Appendix A.1. Recall that the regression coefficient estimate $\hat{\beta}_{jr}$ is the coefficient estimate for variable j and category r, and the probability for customer i to be assigned to category r is π_{ir}. We are looking at the change in probability $\pi_{i,r}$ relative to the probability $\pi_{i,6}$ for category A, hence we can imagine that the regression coefficient estimate for category A is equal to zero, i.e. $\hat{\beta}_{j,6} = 0$ for all j.

In Table 10 we observe the five different regression coefficient estimates for the `Søkeprosess1` variable. This variable relates to if a customer has applied for a loan/credit card. We observe that all of the coefficient estimates are negative, and that $\beta_{5,5}$ is clearly higher than the other estimates. This means that the ratio for category B relative to A, $\pi_{i,5}/\pi_{i,6}$, decreases the least among the five categories. Since the estimate for category A can be imagined as zero and the estimate for category B is relatively close to zero, they form a cluster that stands out from the rest of the estimates. Thus, if we know that the customer has applied for a loan/credit card, the probability for customer i to be assigned to category B and A, i.e. $\pi_{i,5}$ and $\pi_{i,6}$, increases the most of all the categories.

We also observe some significant difference between the groups when it comes to the variable `KRA_Bolig_By_landS`. This variable relates to if the customer lives in an area where the people live far from each other. The estimate for category B and D is clearly lower than the estimates for the other categories. Thus, a customer that lives in a scattered residental area has a reduced probability for being in category B and D.

Moreover, we also see a clear distinction between the categories related to the variable `Medlem`. The coefficient estimates for having this variable set to `Ikke medlem med medlemsbetingelser` or `Medlem` are also presented in Table 10. The first case, i.e. variable $j = 45$, relates to the customers that are non-members with member benefits. We observe that $\beta_{45,1}$ is more negative than the other four estimates. The coefficient estimate is $\beta_{45,1} = 6.931$, which means that if customer i is a non-member with member benefits, the odds relative to the reference category A, i.e. $\pi_{i,1}/\pi_{i,6}$, will decrease by a factor of $\exp(6.931) = 0.00098$. This means that the probability for being in category F, relative to category A, is clearly reduced. Also, since it is the most negative coefficient estimate of this variable, we know that the probability $\pi_{i,1}$ decreases the most among the six probabilities, i.e. π_{ir} where $r = 1, ..., 6$. The latter case, i.e. variable $j = 46$, relates to the customers that are members. We observe that

Variable	**Cat**	**Coef**	**Estimate**	**p–value**
`Søkeprosess1`	F	$\hat{\beta}_{5,1}$	-1.748	$< 2 \cdot 10^{-16}$
	E	$\hat{\beta}_{5,2}$	-1.293	$< 2 \cdot 10^{-16}$
	D	$\hat{\beta}_{5,3}$	-2.309	$< 2 \cdot 10^{-16}$
	C	$\hat{\beta}_{5,4}$	-2.024	$< 2 \cdot 10^{-16}$
	B	$\hat{\beta}_{5,5}$	-0.1320	0.001489
`KRA_Bolig_By_landS`	F	$\hat{\beta}_{34,1}$	-0.0425	0.564444
	E	$\hat{\beta}_{34,2}$	-0.0850	0.132788
	D	$\hat{\beta}_{34,3}$	-0.2311	0.017836
	C	$\hat{\beta}_{34,4}$	-0.0837	0.107769
	B	$\hat{\beta}_{34,5}$	-0.2237	0.000170
`MedlemIkke medlem med medlemsbetingelser`	F	$\hat{\beta}_{45,1}$	-6.931	$7.68 \cdot 10^{-10}$
	E	$\hat{\beta}_{45,2}$	-1.690	0.135106
	D	$\hat{\beta}_{45,3}$	-2.397	0.078874
	C	$\hat{\beta}_{45,4}$	-2.876	0.010624
	B	$\hat{\beta}_{45,5}$	-0.3649	0.772912
`MedlemMedlem`	F	$\hat{\beta}_{46,1}$	-4.134	$< 2 \cdot 10^{-16}$
	E	$\hat{\beta}_{46,2}$	-0.5897	0.233624
	D	$\hat{\beta}_{46,3}$	-0.0335	0.964172
	C	$\hat{\beta}_{46,4}$	0.0793	0.873588
	B	$\hat{\beta}_{46,5}$	0.2104	0.712221
`AvtaleGiro1`	F	$\hat{\beta}_{47,1}$	-8.636	$< 2 \cdot 10^{-16}$
	E	$\hat{\beta}_{47,2}$	-1.902	0.023333
	D	$\hat{\beta}_{47,3}$	-1.173	0.363619
	C	$\hat{\beta}_{47,4}$	-5.208	$1.16 \cdot 10^{-10}$
	B	$\hat{\beta}_{47,5}$	-0.9543	0.285965
`eFaktura1`	F	$\hat{\beta}_{48,1}$	-3.920	$< 2 \cdot 10^{-16}$
	E	$\hat{\beta}_{48,2}$	-3.684	$< 2 \cdot 10^{-16}$
	D	$\hat{\beta}_{48,3}$	-2.946	$< 2 \cdot 10^{-16}$
	C	$\hat{\beta}_{48,4}$	-1.574	$< 2 \cdot 10^{-16}$
	B	$\hat{\beta}_{48,5}$	-1.844	$< 2 \cdot 10^{-16}$

Table 10: This table contains the nominal regression coefficient estimates for six out of 54 variables. Each of the six variables contain estimates that clearly differ from the other categories. Each coefficient estimate (Coef) are listed with their respective category (Cat), estimate and p-value.

$\beta_{46,1}$ is also the most negative of the five estimates. So, if customer i is a member or a non-member with member benefits, the probability for being a customer from category F, i.e. $\pi_{i,1}$, has a clear decrease. Hence, there is a reason to believe that there is a high percentage of non-members in category F.

When it comes to the `AvtaleGiro1` variable, i.e. variable number 47, we observe a clear distinction between $\beta_{47,1}$, $\beta_{47,4}$ and the other coefficient estimates. The estimates for categories C and F is clearly more negative (F more than C), which tells us that the probability for a customer i with direct debit to be categorized in F and C is a significantly reduced (probability for F more reduced than the probability for C). The last variable listed in Table 10 is `eFaktura1`, which is related to customers that have activated electronical billing. We observe that all of these estimates are sufficiently lower than zero. Thus, category A stands out here. The probability for customer i to be categorized in A, i.e. $\pi_{i,6}$, is significantly increased if the customer has electronical billing activated. This is an interesting find, as we observe a clear difference between category A and B. For the most variables, the coefficient estimates for these categories are close to one another.

4.2 Indicator Model

In this section we create an indicator model that indicate customers that are going to be inactive in the future. Before applying variable selection methods to the data set, we look at the effects of each variable in the data set. This is done in Section 4.2.1. Then we will look at the results from the three different variable selection methods in Section 4.2.2. Further, 15 different models are trained on a training set. In Section 4.2.3, we test the performance of each model according to a test set. The performance are evaluated in the form of AUC. Finally, we will interpret the final model, which is going to be the indicator model, in Section 4.2.4.

4.2.1 Interpreting the Full Model

To get a quick overview of the most significant variables, we can use the summary table that was made in Section 3.3.3. We sort the variables according to p-value, since we are looking for variables with significant effects. The lower the p-value, the more significant is the variable. The top ten most significant variables are shown in Table 11. The full summary output is presented in Appendix A.2. Having a positive coefficient estimate will have a positive effect on the linear predictor η_i. This means that the returned probability π_i for customer i to become inactive in the future will increase. Similarly, negative coefficient estimates will have a negative effect on the linear predictor η_i, which again will decrease the probability π_i for customer i to become inactive in the future.

In Table 11 we observe that `SALDO.INNSKUDD` is the *most significant* variable, with a p-value of $3.85 \cdot 10^{-29}$. This coefficient estimate is $-5.24 \cdot 10^{-06}$, so for every one unit increase of this variable will decrease the odds $\pi_i/(1-\pi_i)$ by a factor of $\exp(-5.29 \cdot 10^{-06}) = 0.9999947$. Keep in mind that one unit for this

Variable name	**Estimate**	**Std.Error**	***p*–value**
`SALDO.INNSKUDD`	$-5.29 \cdot 10^{-6}$	$4.54 \cdot 10^{-7}$	$3.85 \cdot 10^{-29}$
`utlan1`	-2.631697	0.2809412	$7.43 \cdot 10^{-21}$
`Lønnsinngang1`	-1.361438	0.1849998	$1.85 \cdot 10^{-13}$
`innskudd1`	-1.234689	0.1755195	$2.00 \cdot 10^{-12}$
`Søkeprosess1`	0.432285	0.0823391	$1.52 \cdot 10^{-07}$
`statusUten aktiv konto`	1.489935	0.2924175	$3.48 \cdot 10^{-7}$
`statusPassiv siste 5 år`	0.851061	0.2288074	$2.00 \cdot 10^{-4}$
`statusAktiv siste år`	0.601313	0.1661946	$2.97 \cdot 10^{-4}$
`innskudd1:Input..kredittkort1`	1.084984	0.3081047	$4.29 \cdot 10^{-4}$
`Fond_avtale1`	-0.635918	0.1806580	$4.32 \cdot 10^{-4}$

Table 11: A list of the ten most significant variables in the full model. The list is obtained by the coefficient table, where the variables are sorted by *p*-value. The coefficient table are provided by the `summary()` function on the glm-object.

variable is 1 NOK, which explains the small change. From another point of view, having a balance of 100000 NOK will decrease the odds $\pi_i/(1-\pi_i)$ by a factor of $\exp(-5.29 \cdot 10^{-06} \cdot 100000) = 0.589$. This variable has a negative effect on the linear predictor η_i, which means that the probability for becoming inactive in the future is reduced. It also makes sense that this variable is one of the most significant. Those who are future active bank customers probably have a balance which is sufficiently greater than zero, while future inactive customers have a balance close to zero. Considering the *p*-value and coefficient estimate, there is a reason to believe that we see this variable as significant in our final model.

The second most significant variable is the `utlan` variable. The coefficient estimate for having the `utlan` variable set to 1, i.e. a customer with loan, is 2.632. This means that the odds, $\pi_i/(1-\pi_i)$, of customer i for becoming inactive decreases by a factor of $\exp(-2.632) = 0.0719$. This variable also makes sense to be significant. Of all the customers with loan, 2.49% had their loan removed and became inactive. Hence, bank customers with loan will most likely owe the bank in the future as well, which counts as using one of their products. A customer that is using one of their products falls into category C, see Table 2. Thus, a customer with loan are using *at least* one product, and will fall into one of the active categories A–C, depending on the usage of other products. Considering the *p*-value and coefficient estimate, there is a reason to believe that we see this variable as significant in our final model as well.

The next variable on the list is the `Lønnsinngang` variable. Having this variable set to 1 relates to the customers that have payroll input. The regression coefficient estimate is 1.361, which means that the odds, $\pi_i/(1-\pi_i)$, of customer i for becoming inactive decreases by a factor of $\exp(1.361) = 0.2564$. Customers usually are most active within the bank where they have their payroll input, so the effect of this variable makes sense. The variable `innskudd` has a similar effect on the linear predictor. This coefficient relates to customers that have a positive balance on their deposit account. This also makes sense, as one could think that a customer would not go inactive unless the balance was close to zero.

We also observe that the `Søkeprosess` variable is significant. If a customer is applying for loan or credit card, the linear predictor η_i for this customer will be increased, together with the probability π_i. This variable is the most significant variable with a positive effect on the linear predictor. This is a bit surprising as the customer is willing to get more products provided by the bank. However, these may be customers that have reached a stage where a mortgage is applicable. The customers have saved up a lot on their accounts in advance, which may explain the activity. Further, they choose to apply for a loan, as well as check the loan conditions with other banks. This may result in inactivity if the application is rejected, or if other banks offer better loan conditions.

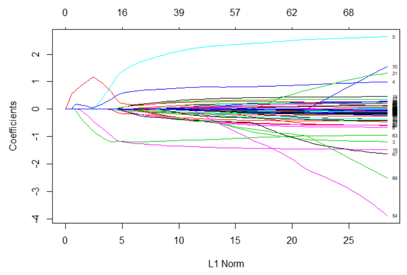
The next three variables are related to the `status` variable, where `Uten aktiv konto` refers to customers without an active account, `Passiv siste 5 år` refers to customers that have been passive the last five years and `Aktiv siste år` refers to customers that have been active the last year. These coefficients are relative to the level `Aktiv siste 5 år`, which refers to the customers that have been active the last 5 years. This explains the positive coefficient estimates. Among the three variables, customers without an active account have the highest increase in the probability and customers that have been active the last year have the lowest increase in the probability, which makes sense.

The interaction variable `innskudd1:Input.aktivitet.kredittkort1` also shows great significance. Customers whose `Input.aktivitet.kredittkort` variable is set to 1 and `innskudd` variable is set to 1, will have an increase of the probability π_i. This is also very interesting. However, this is an interaction term and it can be seen as a correction from the two individual variables. The coefficient estimates for `inskudd1` and `Input.aktivitet.kredittkort1` is -1.234 and -1.328, respectively. Hence, a customer with an active credit card and positive balance on the deposit account will have a total effect of -1.477, which is still negative. This means that this variable has a *reduced* negative effect on the probability π_i.

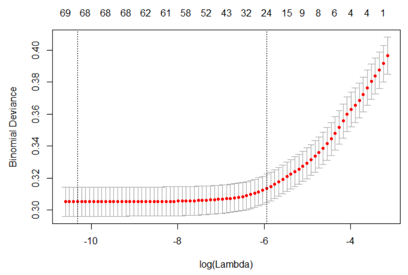
Finally, we have the `Fond_avtale` as the tenth most significant variable. The coefficient estimate for having the `Fond_avtale` variable set to 1, i.e. a customer with fund, is 0.636. This means that the odds, $\pi_i/(1-\pi_i)$, of customer i for becoming inactive decreases by a factor of $\exp(0.636) = 0.529$. The customers with a fund is more connected to the bank, hence this makes sense.

4.2.2 Results from Different Variable Selection Methods

In Section 3.3 we implemented and executed three different methods for model selection. One method was based on the AIC, another method was based on lasso regression and last method was based on variable importance. The returned model for each method are shown in Table 12. The returned model from the `stepAIC()` function has an AIC value of 6357.26, which is lower than the AIC value of full model, that is 6397.96. In addition to this, the returned AIC-model



(a) This plot represents the value of the 71 coefficient estimates, with λ as the parameter. Each line corresponds to a single coefficient estimate. The L1 Norm on the x-axis refers to the amount $\|\beta\|_1$, which increases while λ decreases. The numbers above the plot relates to the number of estimates that have not yet been set to zero.



(b) This plot illustrates the cross-validation error plotted against $\log(\lambda)$. The error is measured in binomial deviance. The value $\log\lambda$ increases as λ increases. The numbers above the plot relates to the number of estimates that have not yet been set to zero.

Figure 6: These plots illustrates the shrinkage of $\boldsymbol{\beta}$ and cross-validation error for different values of the penalty parameter λ.

has only 34 degrees of freedom, compared to the original 71 degrees of freedom in the full model. The number of degrees of freedom is reduced by more than half, yet we have almost the same level of quality.

From the lasso regression in R, we can observe the which variables that are forced to zero given a value of λ. This can be seen in Figure 6a. For example, the cyan line at the very top is referring to variable 5, which is the `innskudd1` variable. Since we have a total of 71 regression coefficients to handle, this plot will visualize 71 lines, which make it hard to distinguish and interpret most of the regression coefficients. Therefore, we will only use this plot only to illustrate that the coefficients are being forced towards zero.

Furthermore, we can plot the cross-validation error from the `cv.glmnet()` function to the corresponding value of λ by simply using the `plot()` function on the returned object. The error is visualized in Figure 6b. Along with the error, which is measured in binomial deviance, we can observe both λ_{min} and λ_{1se} in the figure. We observe that using λ_{min} results in a model with 68 degrees of freedom. That is only a reduction by 3 degrees of freedom from the full model. Using λ_{1se} results in a model with only 24 degrees of freedom, which is far less. This is the reason why we choose to set $\lambda = \lambda_{\text{1se}}$, which in this case is 0.00262. We exclude the variables that have been set to zero in the resulting lasso model.

In Figure 7 we observe the 15 most important variables in random forests. The variable `SALDO.INNSKUDD` is by far the most important variable of them all. The importance is measured in the mean decrease of the Gini index over all M trees in the random forest model. The percentages of all variables sum

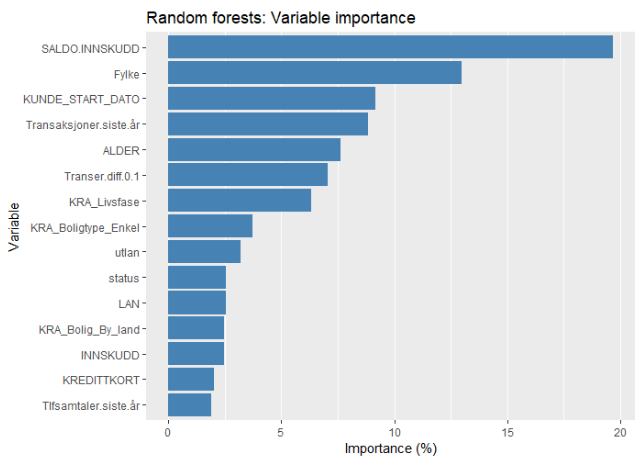


Figure 7: Top 15 most important variables in the random forest model. The importance are measured in mean decrease of the Gini index. The percentages of all variables sum to 100%.

to 100%. The `SALDO.INNSKUDD` variable stands out with almost 10% of the mean decrease in the Gini index. We observe that the top eight variables in this list is either continuous variables or factor with many levels. A factor is a variable with categorical values, where the number of levels corresponds to κ (see Section 3.2.3). Binary variables and factors with few levels are observed to be less important in random forests. When we are making a split in an individual tree structure, we choose the splitting value s that minimize the sum of RSS of the created regions from (2.3). For binary variables, different values of $s \in \{0, 1\}$ will result in the same split. This split will most likely not reduce the Gini index as much as for a continuous variable, where the split is more "customized". Thus, continuous variables and categorical variables with many levels tends to be more important than binary variables in the variable importance plot.

Comparing the returned model for each method help us understand which variables that turns out to be significant on average. Some variables can "get lucky" and come out as significant for one model selection method, but can come out as not significant in an other method. In the same table, we have created a model "Combo", which is a combination model of the all the tree methods. Here we assign the variables that repetitively appear to be significant to the Combo model.

Table 12: Different model selection methods with their returned suggested model. Variables marked with "x" are used in the model. The column AIC refers to the model returned by the `stepAIC()` function. The column Lasso refers to the model returned by the `glmnet()` function. The column VarImp refers to the model with the 20 most important variables in random forests, which can be found in Figure 7. The last column, Combo, includes the variables that repetitively appeared in the returned model from the first three columns.

Nr	Names	AIC	Lasso	VarImp	Combo
2	`status`	x	x	x	x
3	`innskudd`	x	x	x	x
4	`utlan`	x	x	x	x
5	`medlantaker`	x	x		x
9	`BK_KJONN_KODE`			x	
10	`ALDER`	x	x	x	x
11	`KUNDE_START_DATO`	x	x	x	x
14	`LAN`			x	
15	`INNSKUDD`	x	x	x	x
16	`DEBETKORT`				
17	`KREDITTKORT`			x	
18	`NETTBANK`		x	x	x
19	`MOBILBANK`				
20	`BANKID`	x			
21	`SALDO.INNSKUDD`	x	x	x	x
23	`Lønnsinngang`	x	x		x
24	`Søkeprosess`	x	x	x	x
25	`Transaksjoner.siste.år`	x		x	x
30	`Input.aktiv..kredittkort`	x		x	x
31	`Fylke`		x	x	x
34	`Skade_avtale`	x	x	x	x
36	`Nyhetsbrev`				
37	`KRA_Livsfase`		x	x	x
38	`KRA_Bolig_By_land`	x	x	x	x
39	`KRA_Boligtype_Enkel`	x	x	x	x
40	`BIL_I_HUSTAND`	x	x		x
41	`Fond_avtale`	x	x		x
43	`Medlem`				
44	`AvtaleGiro`	x	x		x
45	`eFaktura`				
47	`Tlfsamtaler.siste.år`			x	
48	`Transer.diff.0.1`	x	x	x	x
49	`utlan:0` `Input.aktiv..kredittkort`	x	x		x
50	`innskudd:0` `utlan:0`	x	x		x

Continued on next page

Table 12 – *Continued from previous page*

Nr	Names	AIC	Lasso	VarImp	Combo
51	`innskudd:0` `Input.aktiv..kredittkort`	x			
52	`innskudd:0` `utlan:0` `Input.aktiv..kredittkort:1`	x			

4.2.3 Prediction Methods and Model assessment

We have trained and evaluated 15 different models. We will use the variable combinations found in Table 12 and the full model as different model formulas. Each of these models are then trained by the methods binary logistic regression, random forests or XGBoost. Thus, 5 (different variable combinations) $\times$ 3 (different prediction methods) = 15 different models are trained and evaluated. The number of simulations is set to $N_{\text{sim}} = 100$. The mean test AUC for the different models are presented in Table 13. To get a better overview of the results, the test AUC densities for each model are plotted in Figure 8. In random forests we have set the predictor subset size to $z = \sqrt{k}$ and the number of trees to $M_{\text{RF}} = 200$ trees. When it comes to setting parameters XGBoost, we have used max depth per tree = 8, learning rate $\eta = 0.05$, row and column subsampling parameter $\delta_r = \delta_c = 0.6$, leaf node penalizing factor $\gamma = 2$, weight-smoothing parameter $\nu = 1$ and number of iterations $M_{\text{XGB}} = 200$. The loss function is the binary classification error rate $\mathcal{L}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$, which is the default loss function of `xgb.train()` for classification problems (Chen et al., 2019). The loss is simply the number of wrongly predicted classes divided by the total number of observations n, hence it is highly associated with the 0-1 loss function described in Section 2.5.2. The parameters will be constant for their respective model.

In Figure 8 we observe a big difference when it comes to the prediction accuracy of the three different methods. Binary logistic regression, the green density curves labeled as "GLM", is the method that clearly performs the worst according to AUC. This is as expected, as binary regression only consider linear relations between variables and and the response. Regression in general is great for interpretation of the model, but lacks prediction accuracy. Similarly, tree-based ensemble methods lacks interpretation, but gives remarkable results when it comes to prediction accuracy. Random forests, the red density curves, performs sufficiently better than logistic regression. XGBoost performs the best on average, as all of the five XGBoost models scores a higher test AUC than the five random forest models. It is a clear separation between the regression models and the tree-based models.

It is easier to see differences in numbers than in density plots, thus we will now discuss the results in Table 13. These results origins from the same simulations as used for the density plots in Figure 8. Each model are given a reference

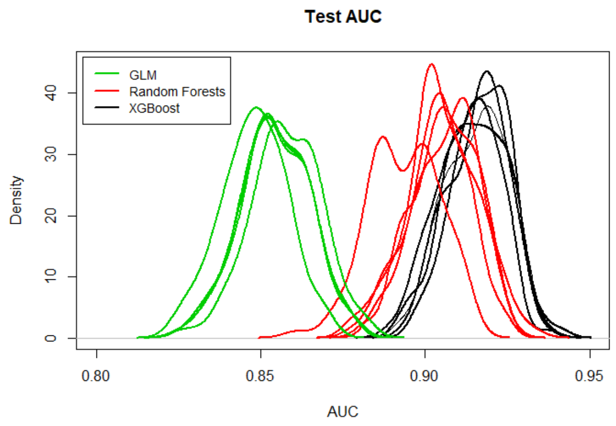


Figure 8: This plot illustrates the test AUC density curves after 100 simulations for the 15 different models. The green density curves represent the binary logistic regression models, the red density curves represent the random forest models and the black density curves represent the XGBoost models.

Method	**Model**	**Model nr**	**Mean**	**Std. dev**
Binary logistic regression	Full	1	0.8543127	0.0104251
	AIC	2	0.8577110	0.0106357
	Lasso	3	0.8543190	0.0104297
	VarImp	4	0.8480242	0.0100768
	Combo	5	0.8544283	0.0104554
Random forests	Full	6	0.9074290	0.0100192
	AIC	7	0.8936977	0.0108110
	Lasso	8	0.9033378	0.0097330
	VarImp	9	0.9061586	0.0095581
	Combo	10	0.9050527	0.0100330
XGBoost	Full	11	0.9174993	0.0086294
	AIC	12	0.9158849	0.0088899
	Lasso	13	0.9129416	0.0092966
	VarImp	14	0.9154367	0.0094171
	Combo	15	0.9156563	0.0094694

Table 13: A list of test AUC results after 100 simulations of the 15 different models. Mean and standard deviance of the test AUC are shown in the two last columns. Each of these columns contains a red and a green number. The red and green number denotes the worst and best value of this column, respectively.

number, for example random forests with the lasso variable combination are referred to as model 8. This table shows the mean and standard deviance of the 100 simulated test AUC values. We want the mean to be as high as possible, and the standard deviance to be as low as possible. The mean corresponds to the performance of the model. The higher the mean, the better is the performance. Standard deviance corresponds to the performance stability and robustness. A model whose performance vary a lot, is not preferred. This can lead to unstable results in projects that is based on this model.

Observing the first five models (1-5), i.e. the logistic regression models, we see that the mean are relatively low. Model 4 resulted in the worst test AUC mean, which is not surprising. This model uses the variables which was found to be most important in random forests. This variable combination is meant for tree-based prediction methods, and this is why this model results in the lowest test AUC mean. Overall, these models have a test AUC mean on around 0.85. I Figure 8, we can clearly see which densities that origins from the logistic regression. Model 2 results in the best regression model, as it has the highest mean and the lowest deviance among the five models.

Moreover, we have the next five models (6-10), which are using random forests. We observe that the test AUC mean is overall around $0.89 - 0.91$. This is far better than the logistic regression models. Model 7 suffers the highest standard deviance of all 15 models, with a value of 0.0108. It also suffers the lowest test AUC mean among the random forest models. Again, this is not surprising. This model uses the variables which are found to be most significant in binary regression models, and is meant for regression models. Model 6 and 9 results in the best random forest models, as they have the highest mean and the lowest deviance among the random forest models, respectively. Since model 9 has a relatively high mean and far less variables, one can argue that this is the preferred random forest model.

Lastly, we have the five final models (11-15), which are trained using XG-Boost. We observe a relatively high overall mean test AUC values, ranging $0.91 - 0.92$. This was expected, due to its reputation of giving state-of-the-art results for a wide range of problems. (Chen and Guestrin, 2016). We will use one of these models as our *final model*, as the models with the overall highest mean and lowest standard deviance are found among these models. That is, model 11 with the highest mean and the lowest standard deviance. It is interesting to see that the model with all the variables included in the model, i.e. model 11, beats the other models when it comes to the mean test AUC. This may be, due to including insignificant variables that make small positive changes. Model 13 suffers the lowest mean and the highest among the XGBoost models, hence this model is outperformed by the others. All in all, we observe that this table is highly related to the test AUC densities in Figure 8, where the green density curve to the very left represents model 4 and the black density curve to the very right represents model 11.

We usually prefer the model that has AUC closest to one, but as mentioned earlier, in applications more weight may be given to sensitivity or to specificity. The idea of the model is to indicate if a bank customer are going to be inactive in

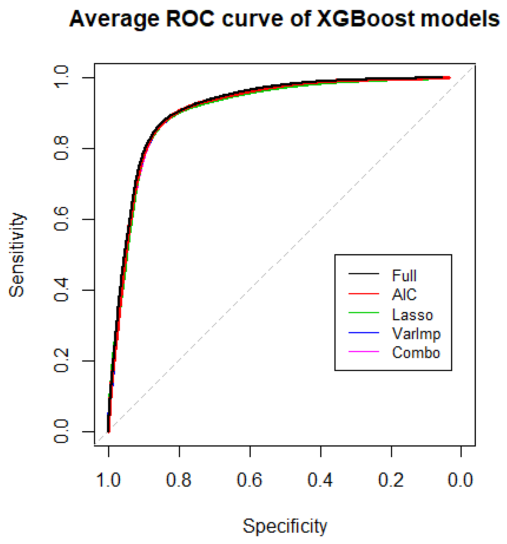


Figure 9: This plot illustrates the average ROC curves after 100 simulations for the 5 different XGBoost models. The black curve represents the full model, the red curve curve represents the AIC model, the green curve represents the lasso model, the blue curve represents the variable importance model and the purple curve represents the combo model.

the future. Indicating a customer that are going to be inactive is more important than indicating a customer that are still going to be active. Thus, we are more interested in sensitivity, i.e. the true positive rate, than specificity, i.e. the true negative rate. Therefore, sensitivity will be more weighted than specificity.

The average ROC curves of $N_{\text{sim}} = 100$ simulations are presented in Figure 9. We observe that the average ROC curves are almost inseparable. The mission here is to see which model that performs best according to the true positive rate. This would be the model represented by the ROC curve whose sensitivity is closest to 1 at any point (vertical direction). This curve would be the black curve representing the full model. This model also turned out to be the model with the highest mean test AUC in Table 13 and the lowest standard deviance. Consequently, we would select model 11 as the final model. However, this model contains all the variables and this ROC curve is very similar to the other four ROC curves. Hence, a simpler model with less variables could be preferred.

In Table 14 we see the number of degrees of freedom, i.e. the number of variables, for each of the five XGBoost models. The simplest model corresponds to the model with the lowest number of degrees of freedom. The maximum number of degrees of freedom is 71, which the full model has. Model 12 has the lowest number of degrees of freedom, 34, which is significantly less. This model also has the second highest mean and the second lowest standard deviance.

Method	Model	Model nr	Mean	Std. dev	Df
XGBoost	Full	11	0.9174993	0.0086294	71
	AIC	12	0.9158849	0.0088899	34
	Lasso	13	0.9129416	0.0092966	57
	VarImp	14	0.9154367	0.0094171	54
	Combo	15	0.9156563	0.0094694	56

Table 14: A list of test AUC results after 100 simulations of the 15 different models. Mean, standard deviance and the degrees of freedom of the test AUC are shown in the three last columns. Each of these columns contains a red and a green number. The red and green number denotes the worst and best value of this column, respectively.

Thus, one could argue that this model would be preferred over the full model. A model with lower degrees of freedom is preferred for simplicity and execution time when training a model. XGBoost handles execution time very well, and a simpler model will not be necessary. Thus, the XGBoost model containing all the variables will be used as the final model. After all, it does perform better than the other 14 models based on AUC.

4.2.4 Interpreting the Indicator Model

We choose the XGBoost model using the all the variables, due to its high mean and low standard deviance on test AUC. This is the model we want to use, to indicate if customer are going from being active to inactive. In Section 4.2.1, we interpreted the full model by using the GLM framework, binary logistic regression. Although, this is not a valid interpretation of the final XGBoost model, it can be used as a simple guide to interpret the effects of each variable on the probabilities, $\boldsymbol{\pi}$. We will interpret the indicator model by calculating the importance of each variable. The variable importance for the indicator model can be found in Figure 10.

Variable importance for an XGBoost model is measured on the amount of Γ_{split}, which is referred to as *gain* by Nielsen (2016) and Chen and Guestrin (2016). The variable importance is presented by percentage of total gain in the model. The value Γ_{split} are used in Algorithm 2, which is the algorithm for XGBoost. The `SALDO.INNSKUDD` variable again is the most important variable, providing almost 35% of the total gain. This variable also turned out to be the most significant variable in logistic regression, Table 11, and in random forests, Figure 7. Thus, it is safe to say that this variable is the most significant of them all. Future active bank customers will probably have a balance which is sufficiently greater than zero, while future inactive customers have a balance close to zero. Thus, the importance of this variable is not surprising.

The second most important variable, providing almost 15% of the gain in the XGBoost model, is the `Transaksjoner.siste.år` variable. This percentage is far less than for the `SALDO.INNSKUDD` variable, observing . The same variable

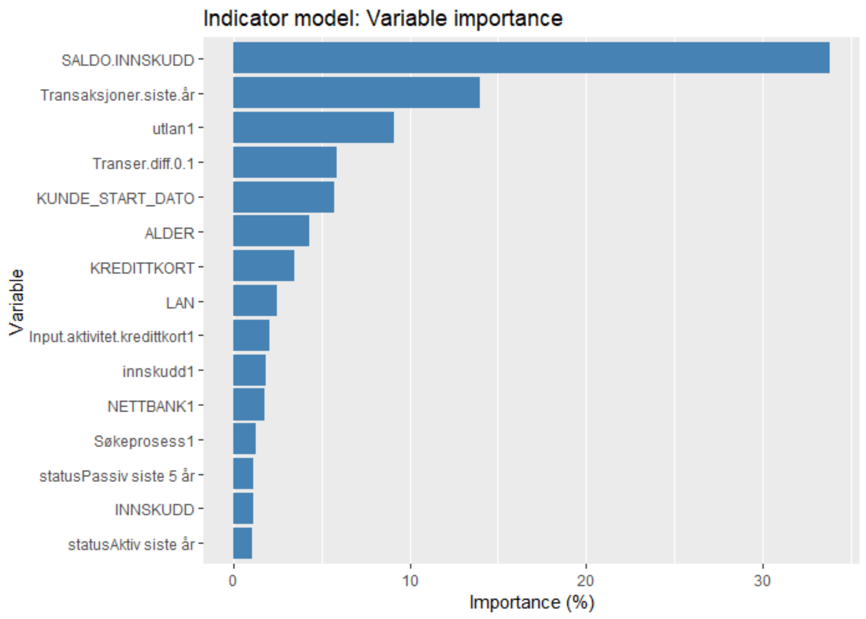


Figure 10: Top 15 most important variables in the indicator model. The importance are measured in gain, Γ_{split}, from (2.15). The percentages of all variables sum to 100%.

was also the fourth most important variable when performing random forests, as observed in Figure 7. This variable is continuous and highly related to the activity of the customer. Hence, it makes sense that this variable is one of the most important variable in the XGBoost model. The third variable is more interesting, as we have a binary variable providing almost 10% of the total gain within the indicator model. This variable, `utlan`, was also found to be significant in the binary regression model, as well as having an increasing effect on the probability of being inactive in the future, if set to 1.

The next variable is `Transer.diff.0.1`, which is also a continuous variable related to activity, similar to the `Transaksjoner.siste.år` variable. This variable relates to the change in number of transactions between the last two years. A positive number shows an increase in number of transactions, while a negative number shows a decrease in the number of transactions. A decrease in activity could mean that this customer will end up in a more inactive category than the current category in the future. This explains the importance of this variable. One can think that, the longer the customer relationship with the bank is, the more loyal is the customer. This may be the reason why the variable `KUNDE_START_DATO` is one of the top five most important variable in the indicator model. From the sixth variable on the variable importance plot in Figure 10 and lower the importance percentage is clearly reduced relative to the number one most important variable.

5 Conclusion and Further Work

We have performed several methods to the customer database, to investigate differences between different types of customers. We performed nominal regression in order to see differences between the coefficient estimates β_{jr}, where r refers to the different customer categories and j is a specific variable. We experienced that customers that have applied for loan/credit card are more likely to be customers from the categories A and B. Similarly, customers that have not applied for loan/credit card are more likely to be customers from the categories C, D, E and F. Also, we observed that the probabilities for being a customer in the categories B and D were clearly reduced if the customer lives in a scattered residential area. Furthermore, we experienced that probability for being in category F is strongly decreased if the customer is a member, a non-member with member benefits or if the customer has activated direct debit. The probability for being in category C is also reduced if the customer has activated direct debit. Finally, the probability for being in category A is significantly increased if the customers have activated electronical billing.

We experienced that the deposit balance of a customer was clearly the most significant variable, when it comes to being active or inactive in the future. Observing the logistic regression coefficient for this variable, it has a negative effect on the probability π_i for being inactive in the future. The number of transactions of the customer and if the customer has a loan, are both also very important factors when it comes to being loyal to the bank.

We performed logistic regression, random forests and XGBoost in order to make the indicator model. We also looked at different variable combinations, to observe the change in performance. The different variable selection methods only had a positive effect in logistic regression model, where the AIC model outperformed the full model. As for the tree-based ensemble methods, the full model outperformed the other models. Looking at the performance of each of the models in the form of AUC, logistic regression was a clear loser, while XGBoost performed best among the tree-based methods. This resulted in the XGBoost model containing all the variables as the indicator model.

So how can this indicator model be used? Customer care and customer insight are two key words to this question. The bank can use the model on their customer database to estimate the probability for each customer to be inactive in the future. The model also gives information about why a given customer have high probability for becoming inactive in the future. Customers with high probability for being inactive can be contacted and informed about the bank product based on the insight about the customers behaviour. Reaching out to the customers before they become inactive can be a huge advantage to help sustain active customers, and will help the bank to provide better customer care.

In addition, by contacting customer with high probability for being inactive, the bank also can learn more about these "types of customers". This insight is valuable for the company, when making new offers or products.

References

Akaike, H. (1974). A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pp. 215–222. Springer.

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

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

Chen, T., T. He, M. Benesty, and V. Khotilovich (2019). Package 'xgboost'. *Cran R.*

Croissant, Y. (2019). Package 'mlogit'.

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

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics New York, NY, USA:.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Hastie, T. and J. Qian (2014). Glmnet vignette. *Retrieve from http://www. web. stanford. edu/˜ hastie/Papers/Glmnet_Vignette. pdf. Accessed September 20*, 2016.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer.

Kullback, S. and R. A. Leibler (1951). On information and sufficiency. *The annals of mathematical statistics 22*(1), 79–86.

Liaw, A. and M. Wiener (2019). Package 'randomforest'. *Cran R.*

Nielsen, D. (2016). Tree boosting with xgboost-why does xgboost win "every" machine learning competition? Master's thesis, NTNU.

Ripley, B., B. Venables, D. M. Bates, K. Hornik, A. Gebhardt, D. Firth, and M. B. Ripley (2013). Package 'mass'. *Cran R.*

Ripley, B. and W. Venables (2016). Package 'nnet'. *R package version 7*, 3–12.

Robin, X., N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, M. Müller, S. Siegert, M. Doering, and M. X. Robin (2019). Package 'proc'.

Yee, T. W. (2019). Package 'vgam'.

A Summary Output

A.1 Nominal Regression

```
                                  Estimate
BK_KJONN_KODEK:1                 1.890e+01
BK_KJONN_KODEK:2                 9.133e+00
BK_KJONN_KODEK:3                 5.185e+00
BK_KJONN_KODEK:4                 1.369e+01
BK_KJONN_KODEK:5                 6.933e+00
BK_KJONN_KODEM:1                 1.879e+01
BK_KJONN_KODEM:2                 9.434e+00
BK_KJONN_KODEM:3                 5.372e+00
BK_KJONN_KODEM:4                 1.393e+01
BK_KJONN_KODEM:5                 7.070e+00
ALDER:1                         -1.455e-02
ALDER:2                         -3.529e-02
ALDER:3                         -3.943e-04
ALDER:4                         -1.408e-02
ALDER:5                         -2.082e-02
KUNDE_START_DATO:1               1.771e-05
KUNDE_START_DATO:2               3.994e-05
KUNDE_START_DATO:3              -6.614e-05
KUNDE_START_DATO:4              -4.257e-05
KUNDE_START_DATO:5              -1.685e-04
Sokeprosess1:1                  -1.748e+00
Sokeprosess1:2                  -1.293e+00
Sokeprosess1:3                  -2.309e+00
Sokeprosess1:4                  -2.024e+00
Sokeprosess1:5                  -1.320e-01
FylkeAust-Agder:1                4.152e-01
FylkeAust-Agder:2                1.243e-01
FylkeAust-Agder:3               -5.410e-02
FylkeAust-Agder:4                2.739e-01
FylkeAust-Agder:5               -1.654e-01
FylkeBuskerud:1                  3.448e-01
FylkeBuskerud:2                  3.502e-02
FylkeBuskerud:3                  3.417e-01
FylkeBuskerud:4                  1.213e-02
FylkeBuskerud:5                 -2.473e-01
FylkeFinnmark:1                  1.795e-01
FylkeFinnmark:2                  1.941e-01
FylkeFinnmark:3                 -2.507e-01
FylkeFinnmark:4                  8.511e-02
FylkeFinnmark:5                 -5.838e-02
FylkeHedmark:1                   3.445e-01
FylkeHedmark:2                   3.100e-02
FylkeHedmark:3                   7.232e-02
FylkeHedmark:4                   6.536e-02
FylkeHedmark:5                  -1.641e-01
FylkeHordaland:1                 1.426e-01
FylkeHordaland:2                -6.515e-02
FylkeHordaland:3                 3.117e-01
FylkeHordaland:4                 1.705e-01
FylkeHordaland:5                -1.849e-01
FylkeMore og Romsdal:1           3.383e-01
FylkeMore og Romsdal:2          -3.259e-02
FylkeMore og Romsdal:3           1.360e-01
FylkeMore og Romsdal:4           1.174e-01
FylkeMore og Romsdal:5          -4.133e-01
FylkeNordland:1                  3.339e-01
FylkeNordland:2                  3.929e-02
FylkeNordland:3                  1.464e-01
FylkeNordland:4                  1.020e-01
FylkeNordland:5                 -2.957e-01
FylkeOppland:1                   3.341e-01
FylkeOppland:2                  -9.469e-03
FylkeOppland:3                   2.749e-01
FylkeOppland:4                   6.508e-02
FylkeOppland:5                  -3.229e-01
FylkeOslo:1                      1.186e-01
FylkeOslo:2                     -1.054e-01
FylkeOslo:3                      6.352e-01
FylkeOslo:4                      1.617e-02
FylkeOslo:5                     -3.326e-04
FylkeRogaland:1                  6.085e-02
FylkeRogaland:2                 -1.261e-01
FylkeRogaland:3                 -1.579e-02
FylkeRogaland:4                  6.117e-02
FylkeRogaland:5                 -3.668e-01
FylkeSogn og Fjordane:1          3.855e-01
FylkeSogn og Fjordane:2          5.633e-02
FylkeSogn og Fjordane:3          8.919e-01
FylkeSogn og Fjordane:4          5.317e-01
FylkeSogn og Fjordane:5         -5.820e-01
FylkeTelemark:1                  6.987e-01
FylkeTelemark:2                  2.949e-01
```

```
FylkeTelemark:3                                      6.854e-01
FylkeTelemark:4                                      3.215e-01
FylkeTelemark:5                                     -1.249e-01
FylkeTroms:1                                         2.084e-01
FylkeTroms:2                                         7.552e-02
FylkeTroms:3                                         1.919e-01
FylkeTroms:4                                         1.069e-01
FylkeTroms:5                                        -4.669e-02
FylkeTrondelag:1                                     6.545e-01
FylkeTrondelag:2                                     2.197e-01
FylkeTrondelag:3                                     5.001e-01
FylkeTrondelag:4                                     2.455e-01
FylkeTrondelag:5                                    -2.362e-01
FylkeVest-Agder:1                                    3.573e-01
FylkeVest-Agder:2                                    1.649e-01
FylkeVest-Agder:3                                    6.158e-01
FylkeVest-Agder:4                                    4.752e-01
FylkeVest-Agder:5                                   -1.916e-01
FylkeVestfold:1                                      1.711e-01
FylkeVestfold:2                                      5.597e-02
FylkeVestfold:3                                     -4.755e-02
FylkeVestfold:4                                     -4.301e-02
FylkeVestfold:5                                     -1.483e-01
Fylkeostfold:1                                       2.085e-01
Fylkeostfold:2                                      -1.580e-01
Fylkeostfold:3                                      -2.864e-01
Fylkeostfold:4                                      -2.340e-01
Fylkeostfold:5                                      -2.465e-01
Skade_avtale1:1                                     -1.178e+00
Skade_avtale1:2                                     -1.052e+00
Skade_avtale1:3                                     -7.957e-01
Skade_avtale1:4                                     -5.818e-01
Skade_avtale1:5                                     -3.019e-01
Nyhetsbrev1:1                                       -3.327e+00
Nyhetsbrev1:2                                       -2.126e+00
Nyhetsbrev1:3                                       -2.848e+00
Nyhetsbrev1:4                                       -3.711e+00
Nyhetsbrev1:5                                       -9.881e-01
KRA_LivsfaseEtablertBarnefamilie:1                   1.226e-02
KRA_LivsfaseEtablertBarnefamilie:2                  -2.790e-01
KRA_LivsfaseEtablertBarnefamilie:3                  -6.605e-02
KRA_LivsfaseEtablertBarnefamilie:4                   2.711e-02
KRA_LivsfaseEtablertBarnefamilie:5                  -1.318e-01
KRA_LivsfaseMiddelaldrendeEnslig:1                  -1.446e-01
KRA_LivsfaseMiddelaldrendeEnslig:2                  -1.679e-01
KRA_LivsfaseMiddelaldrendeEnslig:3                   5.714e-01
KRA_LivsfaseMiddelaldrendeEnslig:4                   2.415e-01
KRA_LivsfaseMiddelaldrendeEnslig:5                   3.606e-02
KRA_LivsfaseMiddelaldrendePar:1                     -3.009e-01
KRA_LivsfaseMiddelaldrendePar:2                     -4.970e-01
KRA_LivsfaseMiddelaldrendePar:3                      2.312e-01
KRA_LivsfaseMiddelaldrendePar:4                      1.543e-01
KRA_LivsfaseMiddelaldrendePar:5                     -2.585e-01
KRA_LivsfaseParUtenBarn:1                           -3.311e-01
KRA_LivsfaseParUtenBarn:2                           -4.517e-01
KRA_LivsfaseParUtenBarn:3                            3.968e-01
KRA_LivsfaseParUtenBarn:4                            1.588e-01
KRA_LivsfaseParUtenBarn:5                           -4.292e-01
KRA_LivsfaseSeniorEnslig:1                          -9.191e-01
KRA_LivsfaseSeniorEnslig:2                          -3.265e-01
KRA_LivsfaseSeniorEnslig:3                           8.043e-01
KRA_LivsfaseSeniorEnslig:4                           4.991e-01
KRA_LivsfaseSeniorEnslig:5                           8.695e-02
KRA_LivsfaseSeniorPar:1                             -5.034e-01
KRA_LivsfaseSeniorPar:2                             -8.921e-01
KRA_LivsfaseSeniorPar:3                              5.027e-01
KRA_LivsfaseSeniorPar:4                              2.671e-01
KRA_LivsfaseSeniorPar:5                             -4.217e-01
KRA_LivsfaseSingel:1                                -6.471e-01
KRA_LivsfaseSingel:2                                -5.801e-01
KRA_LivsfaseSingel:3                                 1.268e-01
KRA_LivsfaseSingel:4                                 5.113e-02
KRA_LivsfaseSingel:5                                -2.608e-01
KRA_LivsfaseSmabarnsfamilie:1                       -2.985e-01
KRA_LivsfaseSmabarnsfamilie:2                       -4.483e-01
KRA_LivsfaseSmabarnsfamilie:3                        2.305e-01
KRA_LivsfaseSmabarnsfamilie:4                       -1.911e-01
KRA_LivsfaseSmabarnsfamilie:5                       -3.938e-01
KRA_LivsfaseUngdomStudent:1                         -1.061e+00
KRA_LivsfaseUngdomStudent:2                         -8.250e-01
KRA_LivsfaseUngdomStudent:3                          1.108e+00
KRA_LivsfaseUngdomStudent:4                          6.920e-01
KRA_LivsfaseUngdomStudent:5                         -5.183e-01
KRA_Bolig_By_landS:1                                -4.252e-02
KRA_Bolig_By_landS:2                                -8.496e-02
KRA_Bolig_By_landS:3                                -2.311e-01
KRA_Bolig_By_landS:4                                -8.368e-02
KRA_Bolig_By_landS:5                                -2.237e-01
KRA_Bolig_By_landT:1                                 1.269e-01
KRA_Bolig_By_landT:2                                -6.237e-03
KRA_Bolig_By_landT:3                                -3.499e-01
```

```
KRA_Bolig_By_landT:4                                                -3.344e-02
KRA_Bolig_By_landT:5                                                -1.243e-01
KRA_Boligtype_EnkelEneboliger:1                                     -1.081e+00
KRA_Boligtype_EnkelEneboliger:2                                     -5.150e-01
KRA_Boligtype_EnkelEneboliger:3                                      3.250e-01
KRA_Boligtype_EnkelEneboliger:4                                     -5.587e-01
KRA_Boligtype_EnkelEneboliger:5                                      7.034e-02
KRA_Boligtype_EnkelFritidsboliger:1                                 -6.661e-01
KRA_Boligtype_EnkelFritidsboliger:2                                  6.867e-02
KRA_Boligtype_EnkelFritidsboliger:3                                  6.516e-01
KRA_Boligtype_EnkelFritidsboliger:4                                 -5.900e-01
KRA_Boligtype_EnkelFritidsboliger:5                                 -4.158e-01
KRA_Boligtype_EnkelGarasje og uthus til bolig:1                     -2.109e+00
KRA_Boligtype_EnkelGarasje og uthus til bolig:2                     -8.830e-01
KRA_Boligtype_EnkelGarasje og uthus til bolig:3                     -1.108e+01
KRA_Boligtype_EnkelGarasje og uthus til bolig:4                     -3.156e-01
KRA_Boligtype_EnkelGarasje og uthus til bolig:5                     -8.253e-01
KRA_Boligtype_EnkelNaringsbygg og andre bygg:1                      -5.393e-01
KRA_Boligtype_EnkelNaringsbygg og andre bygg:2                      -3.110e-01
KRA_Boligtype_EnkelNaringsbygg og andre bygg:3                       4.478e-01
KRA_Boligtype_EnkelNaringsbygg og andre bygg:4                      -3.040e-01
KRA_Boligtype_EnkelNaringsbygg og andre bygg:5                       2.148e-01
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:1             -1.032e+00
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:2             -5.235e-01
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:3              8.940e-02
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:4             -6.760e-01
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:5              7.118e-02
KRA_Boligtype_EnkelStore boligbygg (blokk):1                        -7.752e-01
KRA_Boligtype_EnkelStore boligbygg (blokk):2                        -4.715e-01
KRA_Boligtype_EnkelStore boligbygg (blokk):3                         9.351e-02
KRA_Boligtype_EnkelStore boligbygg (blokk):4                        -5.160e-01
KRA_Boligtype_EnkelStore boligbygg (blokk):5                         5.033e-02
KRA_Boligtype_EnkelTomannsboliger:1                                 -9.021e-01
KRA_Boligtype_EnkelTomannsboliger:2                                 -3.632e-01
KRA_Boligtype_EnkelTomannsboliger:3                                  2.987e-01
KRA_Boligtype_EnkelTomannsboliger:4                                 -4.134e-01
KRA_Boligtype_EnkelTomannsboliger:5                                  2.431e-01
BIL_I_HUSTAND1:1                                                    -2.786e-02
BIL_I_HUSTAND1:2                                                    -1.244e-01
BIL_I_HUSTAND1:3                                                     1.757e-01
BIL_I_HUSTAND1:4                                                    -5.512e-02
BIL_I_HUSTAND1:5                                                    -6.260e-02
Fond_avtale1:1                                                       6.274e-02
Fond_avtale1:2                                                       2.299e-01
Fond_avtale1:3                                                      -3.412e-01
Fond_avtale1:4                                                      -4.193e-01
Fond_avtale1:5                                                      -1.912e-01
MedlemIkke medlem med medlemsbetingelser:1                          -6.931e+00
MedlemIkke medlem med medlemsbetingelser:2                          -1.690e+00
MedlemIkke medlem med medlemsbetingelser:3                          -2.397e+00
MedlemIkke medlem med medlemsbetingelser:4                          -2.876e+00
MedlemIkke medlem med medlemsbetingelser:5                          -3.649e-01
MedlemMedlem:1                                                      -4.134e+00
MedlemMedlem:2                                                      -5.897e-01
MedlemMedlem:3                                                      -3.352e-02
MedlemMedlem:4                                                       7.932e-02
MedlemMedlem:5                                                       2.104e-01
AvtaleGiro1:1                                                       -8.636e+00
AvtaleGiro1:2                                                       -1.902e+00
AvtaleGiro1:3                                                       -1.173e+00
AvtaleGiro1:4                                                       -5.208e+00
AvtaleGiro1:5                                                       -9.543e-01
eFaktura1:1                                                         -3.920e+00
eFaktura1:2                                                         -3.684e+00
eFaktura1:3                                                         -2.946e+00
eFaktura1:4                                                         -1.574e+00
eFaktura1:5                                                         -1.844e+00
Tlfsamtaler.siste.ar:1                                              -1.177e+00
Tlfsamtaler.siste.ar:2                                              -7.965e-01
Tlfsamtaler.siste.ar:3                                              -1.494e+00
Tlfsamtaler.siste.ar:4                                              -3.925e-01
Tlfsamtaler.siste.ar:5                                              -1.207e-01
```

```
                                                    p-value
BK_KJONN_KODEK:1                                   < 2e-16 ***
BK_KJONN_KODEK:2                                  3.77e-09 ***
BK_KJONN_KODEK:3                                  0.018770 *
BK_KJONN_KODEK:4                                   < 2e-16 ***
BK_KJONN_KODEK:5                                  4.63e-05 ***
BK_KJONN_KODEM:1                                   < 2e-16 ***
BK_KJONN_KODEM:2                                  1.16e-09 ***
BK_KJONN_KODEM:3                                  0.014945 *
BK_KJONN_KODEM:4                                   < 2e-16 ***
BK_KJONN_KODEM:5                                  3.30e-05 ***
ALDER:1                                           0.002577 **
ALDER:2                                            < 2e-16 ***
ALDER:3                                           0.953069
ALDER:4                                           4.37e-05 ***
ALDER:5                                           4.44e-08 ***
KUNDE_START_DATO:1                                0.151333
KUNDE_START_DATO:2                                0.000241 ***
KUNDE_START_DATO:3                                2.06e-05 ***
KUNDE_START_DATO:4                                3.47e-06 ***
KUNDE_START_DATO:5                                 < 2e-16 ***
Sokeprosess1:1                                     < 2e-16 ***
Sokeprosess1:2                                     < 2e-16 ***
Sokeprosess1:3                                     < 2e-16 ***
Sokeprosess1:4                                     < 2e-16 ***
Sokeprosess1:5                                    0.001489 **
FylkeAust-Agder:1                                 0.029245 *
FylkeAust-Agder:2                                 0.398948
FylkeAust-Agder:3                                 0.848997
FylkeAust-Agder:4                                 0.041221 *
FylkeAust-Agder:5                                 0.277121
FylkeBuskerud:1                                   0.010204 *
FylkeBuskerud:2                                   0.725192
FylkeBuskerud:3                                   0.057105 .
FylkeBuskerud:4                                   0.894974
FylkeBuskerud:5                                   0.014699 *
FylkeFinnmark:1                                   0.327051
FylkeFinnmark:2                                   0.154425
FylkeFinnmark:3                                   0.415673
FylkeFinnmark:4                                   0.517252
FylkeFinnmark:5                                   0.681757
FylkeHedmark:1                                    0.010542 *
FylkeHedmark:2                                    0.761635
FylkeHedmark:3                                    0.706017
FylkeHedmark:4                                    0.490283
FylkeHedmark:5                                    0.110249
FylkeHordaland:1                                  0.233871
FylkeHordaland:2                                  0.467786
FylkeHordaland:3                                  0.053764 .
FylkeHordaland:4                                  0.038549 *
FylkeHordaland:5                                  0.038364 *
FylkeMore og Romsdal:1                            0.009739 **
FylkeMore og Romsdal:2                            0.749198
FylkeMore og Romsdal:3                            0.452431
FylkeMore og Romsdal:4                            0.207246
FylkeMore og Romsdal:5                            9.82e-05 ***
FylkeNordland:1                                   0.010347 *
FylkeNordland:2                                   0.691550
FylkeNordland:3                                   0.427652
FylkeNordland:4                                   0.267424
FylkeNordland:5                                   0.003901 **
FylkeOppland:1                                    0.016126 *
FylkeOppland:2                                    0.929060
FylkeOppland:3                                    0.139761
FylkeOppland:4                                    0.505916
FylkeOppland:5                                    0.003608 **
FylkeOslo:1                                       0.418250
FylkeOslo:2                                       0.334804
FylkeOslo:3                                       0.000551 ***
FylkeOslo:4                                       0.870495
FylkeOslo:5                                       0.997437
FylkeRogaland:1                                   0.611795
FylkeRogaland:2                                   0.160522
FylkeRogaland:3                                   0.925943
FylkeRogaland:4                                   0.459722
FylkeRogaland:5                                   6.41e-05 ***
FylkeSogn og Fjordane:1                           0.083427 .
FylkeSogn og Fjordane:2                           0.755807
FylkeSogn og Fjordane:3                           0.000760 ***
FylkeSogn og Fjordane:4                           0.001506 **
FylkeSogn og Fjordane:5                           0.008071 **
FylkeTelemark:1                                   1.91e-06 ***
FylkeTelemark:2                                   0.009413 **
FylkeTelemark:3                                   0.000265 ***
FylkeTelemark:4                                   0.001885 **
FylkeTelemark:5                                   0.285656
FylkeTroms:1                                      0.157117
FylkeTroms:2                                      0.489954
FylkeTroms:3                                      0.342115
FylkeTroms:4                                      0.296176
FylkeTroms:5                                      0.669227
```

```
FylkeTrondelag:1                                             2.32e-08 ***
FylkeTrondelag:2                                             0.014770 *
FylkeTrondelag:3                                             0.001882 **
FylkeTrondelag:4                                             0.003155 **
FylkeTrondelag:5                                             0.011197 *
FylkeVest-Agder:1                                            0.051483 .
FylkeVest-Agder:2                                            0.244177
FylkeVest-Agder:3                                            0.008296 **
FylkeVest-Agder:4                                            0.000225 ***
FylkeVest-Agder:5                                            0.193410
FylkeVestfold:1                                              0.171468
FylkeVestfold:2                                              0.542906
FylkeVestfold:3                                              0.788124
FylkeVestfold:4                                              0.611916
FylkeVestfold:5                                              0.097554 .
Fylkeostfold:1                                               0.063943 .
Fylkeostfold:2                                               0.055455 .
Fylkeostfold:3                                               0.088077 .
Fylkeostfold:4                                               0.002160 **
Fylkeostfold:5                                               0.001918 **
Skade_avtale1:1                                               < 2e-16 ***
Skade_avtale1:2                                               < 2e-16 ***
Skade_avtale1:3                                               < 2e-16 ***
Skade_avtale1:4                                               < 2e-16 ***
Skade_avtale1:5                                              6.17e-14 ***
Nyhetsbrev1:1                                                0.001015 **
Nyhetsbrev1:2                                                0.037208 *
Nyhetsbrev1:3                                                0.011264 *
Nyhetsbrev1:4                                                0.000237 ***
Nyhetsbrev1:5                                                0.381570
KRA_LivsfaseEtablertBarnefamilie:1                           0.915269
KRA_LivsfaseEtablertBarnefamilie:2                           0.001272 **
KRA_LivsfaseEtablertBarnefamilie:3                           0.743097
KRA_LivsfaseEtablertBarnefamilie:4                           0.752591
KRA_LivsfaseEtablertBarnefamilie:5                           0.139076
KRA_LivsfaseMiddelaldrendeEnslig:1                           0.279278
KRA_LivsfaseMiddelaldrendeEnslig:2                           0.089486 .
KRA_LivsfaseMiddelaldrendeEnslig:3                           0.006494 **
KRA_LivsfaseMiddelaldrendeEnslig:4                           0.011215 *
KRA_LivsfaseMiddelaldrendeEnslig:5                           0.718414
KRA_LivsfaseMiddelaldrendePar:1                              0.017841 *
KRA_LivsfaseMiddelaldrendePar:2                              1.68e-07 ***
KRA_LivsfaseMiddelaldrendePar:3                              0.256909
KRA_LivsfaseMiddelaldrendePar:4                              0.092496 .
KRA_LivsfaseMiddelaldrendePar:5                              0.008028 **
KRA_LivsfaseParUtenBarn:1                                    0.022612 *
KRA_LivsfaseParUtenBarn:2                                    3.88e-05 ***
KRA_LivsfaseParUtenBarn:3                                    0.091727 .
KRA_LivsfaseParUtenBarn:4                                    0.143492
KRA_LivsfaseParUtenBarn:5                                    0.000228 ***
KRA_LivsfaseSeniorEnslig:1                                   5.63e-06 ***
KRA_LivsfaseSeniorEnslig:2                                   0.029199 *
KRA_LivsfaseSeniorEnslig:3                                   0.003402 **
KRA_LivsfaseSeniorEnslig:4                                   0.000204 ***
KRA_LivsfaseSeniorEnslig:5                                   0.552904
KRA_LivsfaseSeniorPar:1                                      0.004133 **
KRA_LivsfaseSeniorPar:2                                      3.54e-11 ***
KRA_LivsfaseSeniorPar:3                                      0.048312 *
KRA_LivsfaseSeniorPar:4                                      0.029244 *
KRA_LivsfaseSeniorPar:5                                      0.001945 **
KRA_LivsfaseSingel:1                                         1.94e-05 ***
KRA_LivsfaseSingel:2                                         1.15e-07 ***
KRA_LivsfaseSingel:3                                         0.605719
KRA_LivsfaseSingel:4                                         0.636035
KRA_LivsfaseSingel:5                                         0.020903 *
KRA_LivsfaseSmabarnsfamilie:1                                0.017130 *
KRA_LivsfaseSmabarnsfamilie:2                                1.68e-06 ***
KRA_LivsfaseSmabarnsfamilie:3                                0.272782
KRA_LivsfaseSmabarnsfamilie:4                                0.042406 *
KRA_LivsfaseSmabarnsfamilie:5                                5.76e-05 ***
KRA_LivsfaseUngdomStudent:1                                  0.000758 ***
KRA_LivsfaseUngdomStudent:2                                  0.001444 **
KRA_LivsfaseUngdomStudent:3                                  0.004429 **
KRA_LivsfaseUngdomStudent:4                                  0.006067 **
KRA_LivsfaseUngdomStudent:5                                  0.073071 .
KRA_Bolig_By_landS:1                                         0.564444
KRA_Bolig_By_landS:2                                         0.132788
KRA_Bolig_By_landS:3                                         0.017836 *
KRA_Bolig_By_landS:4                                         0.107769
KRA_Bolig_By_landS:5                                         0.000170 ***
KRA_Bolig_By_landT:1                                         0.048736 *
KRA_Bolig_By_landT:2                                         0.900943
KRA_Bolig_By_landT:3                                         0.000116 ***
KRA_Bolig_By_landT:4                                         0.469792
KRA_Bolig_By_landT:5                                         0.017052 *
KRA_Boligtype_EnkelEneboliger:1                              0.130848
KRA_Boligtype_EnkelEneboliger:2                              0.389963
KRA_Boligtype_EnkelEneboliger:3                              0.777192
KRA_Boligtype_EnkelEneboliger:4                              0.320392
KRA_Boligtype_EnkelEneboliger:5                              0.916400
KRA_Boligtype_EnkelFritidsboliger:1                          0.426737
```

```
KRA_Boligtype_EnkelFritidsboliger:2                     0.919041
KRA_Boligtype_EnkelFritidsboliger:3                     0.612033
KRA_Boligtype_EnkelFritidsboliger:4                     0.352628
KRA_Boligtype_EnkelFritidsboliger:5                     0.588785
KRA_Boligtype_EnkelGarasje og uthus til bolig:1         0.126663
KRA_Boligtype_EnkelGarasje og uthus til bolig:2         0.419395
KRA_Boligtype_EnkelGarasje og uthus til bolig:3               NA
KRA_Boligtype_EnkelGarasje og uthus til bolig:4         0.735675
KRA_Boligtype_EnkelGarasje og uthus til bolig:5         0.540947
KRA_Boligtype_EnkelNaringsbygg og andre bygg:1          0.470215
KRA_Boligtype_EnkelNaringsbygg og andre bygg:2          0.618206
KRA_Boligtype_EnkelNaringsbygg og andre bygg:3          0.704823
KRA_Boligtype_EnkelNaringsbygg og andre bygg:4          0.603383
KRA_Boligtype_EnkelNaringsbygg og andre bygg:5          0.756949
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:1 0.150577
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:2 0.383010
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:3 0.938082
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:4 0.230015
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus:5 0.915518
KRA_Boligtype_EnkelStore boligbygg (blokk):1            0.279519
KRA_Boligtype_EnkelStore boligbygg (blokk):2            0.431676
KRA_Boligtype_EnkelStore boligbygg (blokk):3            0.935159
KRA_Boligtype_EnkelStore boligbygg (blokk):4            0.359172
KRA_Boligtype_EnkelStore boligbygg (blokk):5            0.940167
KRA_Boligtype_EnkelTomannsboliger:1                     0.209757
KRA_Boligtype_EnkelTomannsboliger:2                     0.545898
KRA_Boligtype_EnkelTomannsboliger:3                     0.795586
KRA_Boligtype_EnkelTomannsboliger:4                     0.463949
KRA_Boligtype_EnkelTomannsboliger:5                     0.717690
BIL_I_HUSTAND1:1                                        0.752042
BIL_I_HUSTAND1:2                                        0.067013 .
BIL_I_HUSTAND1:3                                        0.157900
BIL_I_HUSTAND1:4                                        0.387765
BIL_I_HUSTAND1:5                                        0.367526
Fond_avtale1:1                                          0.471401
Fond_avtale1:2                                          0.000843 ***
Fond_avtale1:3                                          0.001168 **
Fond_avtale1:4                                          3.65e-12 ***
Fond_avtale1:5                                          0.004916 **
MedlemIkke medlem med medlemsbetingelser:1              7.68e-10 ***
MedlemIkke medlem med medlemsbetingelser:2              0.135106
MedlemIkke medlem med medlemsbetingelser:3              0.078874 .
MedlemIkke medlem med medlemsbetingelser:4              0.010624 *
MedlemIkke medlem med medlemsbetingelser:5              0.772912
MedlemMedlem:1                                           < 2e-16 ***
MedlemMedlem:2                                          0.233624
MedlemMedlem:3                                          0.964172
MedlemMedlem:4                                          0.873588
MedlemMedlem:5                                          0.712221
AvtaleGiro1:1                                            < 2e-16 ***
AvtaleGiro1:2                                           0.023333 *
AvtaleGiro1:3                                           0.363619
AvtaleGiro1:4                                           1.16e-10 ***
AvtaleGiro1:5                                           0.285965
eFaktura1:1                                              < 2e-16 ***
eFaktura1:2                                              < 2e-16 ***
eFaktura1:3                                              < 2e-16 ***
eFaktura1:4                                              < 2e-16 ***
eFaktura1:5                                              < 2e-16 ***
Tlfsamtaler.siste.ar:1                                   < 2e-16 ***
Tlfsamtaler.siste.ar:2                                   < 2e-16 ***
Tlfsamtaler.siste.ar:3                                   < 2e-16 ***
Tlfsamtaler.siste.ar:4                                   < 2e-16 ***
Tlfsamtaler.siste.ar:5                                   < 2e-16 ***
```

A.2 Binary Logistic Regression - Full Model

```
                                                                  Estimate
(Intercept)                                                      8.016e-01
statusAktiv siste ar                                             6.013e-01
statusPassiv siste 5 ar                                          8.511e-01
statusUten aktiv konto                                           1.490e+00
innskudd1                                                       -1.235e+00
utlan1                                                          -2.632e+00
medlantakerIkke medlantaker                                      5.759e-01
BK_KJONN_KODEM                                                   1.778e-01
ALDER                                                           -1.009e-02
KUNDE_START_DATO                                                -3.626e-05
LAN                                                              3.272e-02
INNSKUDD                                                        -1.029e-01
DEBETKORT                                                       -1.951e-02
KREDITTKORT                                                     -1.190e-01
NETTBANK1                                                       -5.175e-01
MOBILBANK1                                                      -1.053e-02
BANKID                                                           2.624e-01
SALDO.INNSKUDD                                                  -5.294e-06
Lonnsinngang1                                                   -1.361e+00
Sokeprosess1                                                     4.323e-01
Transaksjoner.siste.ar                                          -2.857e-04
Input.aktivitet.kredittkort1                                    -1.328e+00
FylkeAust-Agder                                                  2.845e-01
FylkeBuskerud                                                    6.003e-02
FylkeFinnmark                                                    2.514e-01
FylkeHedmark                                                     7.558e-02
FylkeHordaland                                                  -5.654e-02
FylkeMore og Romsdal                                             1.966e-01
FylkeNordland                                                    7.991e-02
FylkeOppland                                                     1.224e-01
FylkeOslo                                                        1.271e-01
FylkeRogaland                                                   -2.369e-03
FylkeSogn og Fjordane                                           -3.232e-01
FylkeTelemark                                                   -5.982e-02
FylkeTroms                                                       1.043e-01
FylkeTrondelag                                                   2.791e-01
FylkeVest-Agder                                                 -3.429e-01
FylkeVestfold                                                    1.125e-01
Fylkeostfold                                                     1.078e-01
Skade_avtale1                                                   -2.819e-01
Nyhetsbrev1                                                     -2.269e-02
KRA_LivsfaseEtablertBarnefamilie                                -3.266e-01
KRA_LivsfaseMiddelaldrendeEnslig                                -5.811e-02
KRA_LivsfaseMiddelaldrendePar                                   -1.898e-01
KRA_LivsfaseParUtenBarn                                         -1.356e-01
KRA_LivsfaseSeniorEnslig                                        -3.742e-01
KRA_LivsfaseSeniorPar                                           -3.501e-01
KRA_LivsfaseSingel                                              -5.212e-01
KRA_LivsfaseSmabarnsfamilie                                     -1.608e-01
KRA_LivsfaseUngdomStudent                                       -7.178e-01
KRA_Bolig_By_landS                                              -2.094e-01
KRA_Bolig_By_landT                                              -5.883e-02
KRA_Boligtype_EnkelEneboliger                                   -8.557e-02
KRA_Boligtype_EnkelFritidsboliger                                5.060e-01
KRA_Boligtype_EnkelGarasje og uthus til bolig                   -1.252e+01
KRA_Boligtype_EnkelNaringsbygg og andre bygg                     7.349e-02
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus            1.664e-01
KRA_Boligtype_EnkelStore boligbygg (blokk)                       2.381e-01
KRA_Boligtype_EnkelTomannsboliger                               -3.007e-01
BIL_I_HUSTAND1                                                   1.970e-01
Fond_avtale1                                                    -6.359e-01
MedlemIkke medlem med medlemsbetingelser                        -2.185e-01
MedlemMedlem                                                    -1.913e-01
AvtaleGiro1                                                     -6.474e-01
eFaktura1                                                        2.955e-03
Tlfsamtaler.siste.ar                                             1.064e-02
Transer.diff.0.1                                                -4.359e-04
utlan1:Input.aktivitet.kredittkort1                             -1.107e+01
innskudd1:utlan1                                                -2.981e-01
innskudd1:Input.aktivitet.kredittkort1                           1.085e+00
innskudd1:utlan1:Input.aktivitet.kredittkort1                    1.282e+01
```

```
                                                          Std. Error
(Intercept)                                               1.234e+00
statusAktiv siste ar                                      1.662e-01
statusPassiv siste 5 ar                                   2.288e-01
statusUten aktiv konto                                    2.924e-01
innskudd1                                                 1.755e-01
utlan1                                                    2.809e-01
medlantakerIkke medlantaker                               1.956e-01
BK_KJONN_KODEM                                            7.429e-02
ALDER                                                     6.650e-03
KUNDE_START_DATO                                          1.919e-05
LAN                                                       1.166e-01
INNSKUDD                                                  4.488e-02
DEBETKORT                                                 9.073e-02
KREDITTKORT                                               2.070e-01
NETTBANK1                                                 5.411e-01
MOBILBANK1                                                9.837e-02
BANKID                                                    1.350e-01
SALDO.INNSKUDD                                            4.543e-07
Lonnsinngang1                                             1.850e-01
Sokeprosess1                                              8.234e-02
Transaksjoner.siste.ar                                    1.815e-04
Input.aktivitet.kredittkort1                              4.869e-01
FylkeAust-Agder                                           2.475e-01
FylkeBuskerud                                             1.921e-01
FylkeFinnmark                                             2.610e-01
FylkeHedmark                                              1.880e-01
FylkeHordaland                                            1.655e-01
FylkeMore og Romsdal                                      1.745e-01
FylkeNordland                                             1.876e-01
FylkeOppland                                              1.977e-01
FylkeOslo                                                 1.874e-01
FylkeRogaland                                             1.702e-01
FylkeSogn og Fjordane                                     3.231e-01
FylkeTelemark                                             2.082e-01
FylkeTroms                                                1.962e-01
FylkeTrondelag                                            1.604e-01
FylkeVest-Agder                                           2.562e-01
FylkeVestfold                                             1.682e-01
Fylkeostfold                                              1.623e-01
Skade_avtale1                                             8.251e-02
Nyhetsbrev1                                               1.062e-01
KRA_LivsfaseEtablertBarnefamilie                          1.621e-01
KRA_LivsfaseMiddelaldrendeEnslig                          1.749e-01
KRA_LivsfaseMiddelaldrendePar                             1.707e-01
KRA_LivsfaseParUtenBarn                                   1.909e-01
KRA_LivsfaseSeniorEnslig                                  2.676e-01
KRA_LivsfaseSeniorPar                                     2.435e-01
KRA_LivsfaseSingel                                        2.024e-01
KRA_LivsfaseSmabarnsfamilie                               1.693e-01
KRA_LivsfaseUngdomStudent                                 3.337e-01
KRA_Bolig_By_landS                                        1.072e-01
KRA_Bolig_By_landT                                        9.096e-02
KRA_Boligtype_EnkelEneboliger                             1.060e+00
KRA_Boligtype_EnkelFritidsboliger                         1.191e+00
KRA_Boligtype_EnkelGarasje og uthus til bolig             4.083e+02
KRA_Boligtype_EnkelNaringsbygg og andre bygg              1.092e+00
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus     1.063e+00
KRA_Boligtype_EnkelStore boligbygg (blokk)                1.061e+00
KRA_Boligtype_EnkelTomannsboliger                         1.066e+00
BIL_I_HUSTAND1                                            1.170e-01
Fond_avtale1                                              1.807e-01
MedlemIkke medlem med medlemsbetingelser                  3.513e-01
MedlemMedlem                                              2.389e-01
AvtaleGiro1                                               5.336e-01
eFaktura1                                                 1.186e-01
Tlfsamtaler.siste.ar                                      3.563e-02
Transer.diff.0.1                                          2.159e-04
utlan1:Input.aktivitet.kredittkort1                       2.209e+02
innskudd1:utlan1                                          3.139e-01
innskudd1:Input.aktivitet.kredittkort1                    3.081e-01
innskudd1:utlan1:Input.aktivitet.kredittkort1             2.209e+02
```

```
                                                                 p-value
(Intercept)                                                      0.516026
statusAktiv siste ar                                             0.000297 ***
statusPassiv siste 5 ar                                          0.000200 ***
statusUten aktiv konto                                           3.48e-07 ***
innskudd1                                                        2.00e-12 ***
utlan1                                                            < 2e-16 ***
medlantakerIkke medlantaker                                      0.003230 **
BK_KJONN_KODEM                                                   0.016678 *
ALDER                                                            0.129177
KUNDE_START_DATO                                                 0.058783 .
LAN                                                              0.779016
INNSKUDD                                                         0.021848 *
DEBETKORT                                                        0.829724
KREDITTKORT                                                      0.565468
NETTBANK1                                                        0.338911
MOBILBANK1                                                       0.914723
BANKID                                                           0.051935 .
SALDO.INNSKUDD                                                    < 2e-16 ***
Lonnsinngang1                                                    1.85e-13 ***
Sokeprosess1                                                     1.52e-07 ***
Transaksjoner.siste.ar                                           0.115412
Input.aktivitet.kredittkort1                                     0.006391 **
FylkeAust-Agder                                                  0.250445
FylkeBuskerud                                                    0.754629
FylkeFinnmark                                                    0.335491
FylkeHedmark                                                     0.687654
FylkeHordaland                                                   0.732640
FylkeMore og Romsdal                                             0.260042
FylkeNordland                                                    0.670118
FylkeOppland                                                     0.535742
FylkeOslo                                                        0.497887
FylkeRogaland                                                    0.988897
FylkeSogn og Fjordane                                            0.317218
FylkeTelemark                                                    0.773802
FylkeTroms                                                       0.594895
FylkeTrondelag                                                   0.081794 .
FylkeVest-Agder                                                  0.180660
FylkeVestfold                                                    0.503574
Fylkeostfold                                                     0.506693
Skade_avtale1                                                    0.000635 ***
Nyhetsbrev1                                                      0.830864
KRA_LivsfaseEtablertBarnefamilie                                 0.043859 *
KRA_LivsfaseMiddelaldrendeEnslig                                 0.739696
KRA_LivsfaseMiddelaldrendePar                                    0.266218
KRA_LivsfaseParUtenBarn                                          0.477393
KRA_LivsfaseSeniorEnslig                                         0.162054
KRA_LivsfaseSeniorPar                                            0.150491
KRA_LivsfaseSingel                                               0.010030 *
KRA_LivsfaseSmabarnsfamilie                                      0.342257
KRA_LivsfaseUngdomStudent                                        0.031483 *
KRA_Bolig_By_landS                                               0.050648 .
KRA_Bolig_By_landT                                               0.517808
KRA_Boligtype_EnkelEneboliger                                    0.935686
KRA_Boligtype_EnkelFritidsboliger                                0.670935
KRA_Boligtype_EnkelGarasje og uthus til bolig                    0.975529
KRA_Boligtype_EnkelNaringsbygg og andre bygg                     0.946334
KRA_Boligtype_EnkelRekkehus, kjedehus og andre smahus            0.875540
KRA_Boligtype_EnkelStore boligbygg (blokk)                       0.822509
KRA_Boligtype_EnkelTomannsboliger                                0.777811
BIL_I_HUSTAND1                                                   0.092210 .
Fond_avtale1                                                     0.000432 ***
MedlemIkke medlem med medlemsbetingelser                         0.534000
MedlemMedlem                                                     0.423222
AvtaleGiro1                                                      0.225037
eFaktura1                                                        0.980115
Tlfsamtaler.siste.ar                                             0.765252
Transer.diff.0.1                                                 0.043509 *
utlan1:Input.aktivitet.kredittkort1                              0.960027
innskudd1:utlan1                                                 0.342305
innskudd1:Input.aktivitet.kredittkort1                           0.000429 ***
innskudd1:utlan1:Input.aktivitet.kredittkort1                    0.953735
```