Stockholms universitet

Predicting bank marketing success during a period characterized by financial instability

Felix Seo

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2023:5
Matematisk statistik
Januari 2023

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematiska institutionen

Mathematical Statistics
Stockholm University
Bachelor Thesis **2023:5**
http://www.math.su.se

Predicting bank marketing success during a period characterized by financial instability

Felix Seo*

June 2023

Abstract

In this thesis we use two different models for predicting the success of bank telemarketing campaigns to sell long-term deposits. The data set is related to a Portuguese bank which were collected from 2008 to 2010, hence effects of the financial crisis are included. The data set consists of 20 predictor variables and a response which represents success or failure of selling a long-term deposit to the client. An undersampling of the data set was needed since it was imbalanced with mostly failures as realizations. The models in question are the group lasso for logistic regression and classification trees. We mainly use the prediction error on a test data set (20 performance, but the interpretability of the model is also an important property. Both models acquire similar results of 72classification tree presents the best overall results. The classification tree reveals several key variables in selling long-term deposits (e.g. Euribor rate) while the group lasso can not validate important features. While the group lasso has several interesting and useful properties, further research on the method is needed to decrease its limitations.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: felix.seo1999@gmail.com. Supervisor: Taras Bodnar, Lina Palmborg, Dongni Zhang.

Predicting bank marketing success during a period characterized by financial instability

Felix Seo

January 22, 2023

Contents

Abstract

In this thesis we use two different models for predicting the success of bank telemarketing campaigns to sell long-term deposits. The data set is related to a Portuguese bank which were collected from 2008 to 2010, hence effects of the financial crisis are included. The data set consists of 20 predictor variables and a response which represents success or failure of selling a long-term deposit to the client. An under-sampling of the data set was needed since it was imbalanced with mostly failures as realizations. The models in question are the group lasso for logistic regression and classification trees. We mainly use the prediction error on a test data set (20 % of data) as a metric of performance, but the interpretability of the model is also an important property. Both models acquire similar results of 72% accuracy but the classification tree presents the best overall results. The classification tree reveals several key variables in selling long-term deposits (e.g. Euribor rate) while the group lasso can not validate important features. While the group lasso has several interesting and useful properties, further research on the method is needed to decrease its limitations.

Foreword

This work constitutes a Bachelor's thesis of 15 ECTS in Mathematical Statistics at Stockholm University. I would like to express my gratitude to my supervisors Taras Bodnar, Lina Palmborg and Dongni Zhang for the discussions and the encouraging feedback that amounted to this thesis.

1 Introduction

A common strategy to enhance business is to use marketing selling campaigns. One widely used strategy is direct marketing where contact centers are organized to directly target specific clients by i.e. outbound phone calls. These call centers are used since it simplifies the operational aspect of the marketing campaign. Since it is simpler than before to gather and store data of clients a new task becomes increasingly more intriguing, namely focusing on to maximize the lifetime of a customer and finding new potential customers with greater probability of success [1, Moro et al. 2014].

The data set is related to a marketing campaign of a Portuguese bank which was conducted with phone calls to potential clients during 2008 to 2013. It should be noted that a subset of the data set studied in [1, Moro et al. 2014] including only 20 features and realizations during 2008 to 2010 is used in this thesis. A subset is used since it is the only accessible data. In the data set studied by [1, Moro et al. 2014], they use different data mining approaches to predict the success of selling bank long-term deposits. The data mining models used are logistic regression, classification trees, support vector machines and neural networks. They also needed to do a feature selection beforehand since their data consist of 150 features, ending up with 22 features used in the model fitting. One of the main problems is that the number of features in the data set which is received after the feature selection phase is still quite large and contain a variety of different data types.

The data studied in this thesis also have quite many features and also contain many different data types. Much focus should hence be put towards choosing an appropriate model for the data at hand. In this thesis we investigate one new type of model which is a more general lasso model, namely the group lasso for logistic regression, and also conduct a more rigorous construction of the classification tree. The classification tree in [1, Moro et al. 2014] is constructed by using default parameter values in the R package that fits the classification tree and no further discussion is presented. Focus is to examine the predicative power of the two models, but also interpretability.

In the classical linear regression setting the coefficient estimates are obtained by minimizing the squared loss which often gives nonzero estimates. For large data sets this is a problem due to the fact that all features are then included and the interpretability of the model is then impaired. The lasso [2, Tibshirani 1996] is a popular method to remedy this problem by restricting the coefficient estimates and thus possesses a subset selection property. Although in general the lasso works quite well it does not cover all types of data, e.g. when the variables possess a group structure it can not adequately handle this property.

The group lasso [3, Yuan and Lin 2006] is an extension of the regular lasso which takes group structure in consideration. Yuan and Lin (2006) motivated the group lasso by the multifactor analysis-of-variance (ANOVA) problem and the additive model. In the ANOVA, each factor could be expressed by a group of dummy variables. Often it is desirable to find the main effects and then deleting irrelevant factors which would amount to deleting the group of corresponding dummy variables. In the additive model example they use polynomial or nonparametric components to in both cases express these components as a linear combination of basis functions of the original variables. Removing an unimportant component in the additive model would translate into removing groups of basis functions.

After the introduction of the group lasso there has been some further research regarding the group lasso. The group lasso has been extended for use in logistic regression models [4, Meier et al. 2008]. But both aforementioned papers derive solutions for the group lasso under the assumption of group-wise orthonormality. Although one can often transform the group data beforehand to meet the orthonormality condition, this may not give solution to the original problem [5, Friedman et al. 2010]. However, recent work [6, Yang and Zou 2015] has proposed an algorithm that solves the group lasso problem without the assumption of group-wise orthonormality.

Classification and regression trees [7, Brieman et al. 1984] is a popular tree-based method. It partitions the feature space into rectangles and then fits each region with a constant to serve as the estimated prediction. The method is quite simple but powerful and has been applied in many areas and seen extensions, like random forests and gradient boosting trees. Trees have nice properties that make them often a natural first choice for regression or classification tasks. The way a tree is constructed makes it able to handle data of mixed type (categorical, numerical) and trees are also robust to correlated variables.

The thesis is organized as follows: in Section 2 we present the theory of the models used; in Section 3 we present the data set and conduct some preliminary explanatory analysis; Section 4 describes and analyses the results of the models; finally, Section 5 contains the conclusion of the results and further improvements.

2 Theory

2.1 Logistic regression

Assume that $y_1, y_2, \ldots, y_n$ is a sample from $Y_1, Y_2, \ldots, Y_n$ where all Y_i are independent and normally distributed random variables with mean μ_i and mutual variance σ^2. Also assume that $Y_i = \mu_i + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, then the linear model assumes the form

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i,$$

where β_0, β_j are some unknown parameters and $x_{i1}, x_{i2}, \ldots, x_{ip}$ are some known values (predictors) for $j = 1, \ldots, p$ and $i = 1, \ldots, n$. The model can be expressed in vector notation for convenience with $\beta \in \mathbb{R}^p$ containing all β_j and $X_i = (x_{i1}, \ldots, x_{ip})$ a vector of the x_{ij} such that $y_i = \beta_0 + X_i\beta + \epsilon_i$. Then with the linear model we try to estimate the response y_i by [8, Sundberg 2020, ch. 2.1]

$$\mathrm{E}(Y_i | X = X_i) = \mu_i = \beta_0 + X_i\beta.$$

The linear regression model is just a special case of a much more general model. Consider that we want to approximate $g(\mathrm{E}(Y_i|X = X_i)) = g(\mu_i)$, where $g : \mathbb{R} \to \mathbb{R}$ is a strictly monotonic function, using the linear model

$$g(\mu_i) = \beta_0 + X_i\beta.$$

Choosing $g(\mu_i) = \mu_i$ assuming that the response is Gaussian gives us the standard linear regression model. The function g is called the link function. If we are interested in modeling a binary response $Y \in \{0, 1\}$ then often the linear logistic regression is used with $\mu_i = P(Y = 1|X = X_i)$ and $g(\mu_i) = \log(\mu_i/(1 - \mu_i))$ assuming Bernoulli distributed responses. That is we model the log-ratio of the conditional probabilities

$$\log \frac{P(Y = 1|X = X_i)}{P(Y = 0|X = X_i)} = \beta_0 + X_i\beta. \quad (2.1)$$

Which becomes

$$P(Y = 1|X = X_i) = \frac{e^{\beta_0 + X_i\beta}}{1 + e^{\beta_0 + X_i\beta}},$$

and the predicted probabilities will be in (0,1) [9, Hastie et al. 2016, ch. 3.1].

2.2 Lasso

We know that for a sample $y_1, \ldots, y_n$ the linear regression model estimates the response y_i with $\hat{\mu}_i = \hat{\beta}_0 + X_i\hat{\beta}$. The parameters $\hat{\beta}_0, \hat{\beta}$ are derived by minimizing the squared loss:

$$\min_{\beta_0,\beta} \left(\frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - X_i\beta)^2 \right).$$

It should be noted that under the normality assumption of y_i, minimizing the squared loss is equivalent to minimizing the negative log-likelihood in the linear regression model. The estimated parameters often become non-zero which in turn could make it quite difficult to interpret the model if we have a lot of predictors [9, Hastie et al. 2016, ch. 1]. There often is a smaller subset of the predictors that explain most of the response and by omitting some predictors or shrinking their effects, one may get a higher prediction accuracy in comparison with the full model [9, Hastie et al. 2016, ch. 2.1]. To solve this problem one can introduce the ℓ_1 constraint to the squared loss (called the lasso). So the optimization problem instead becomes

$$\min_{\beta_0,\beta} \left(\frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - X_i\beta)^2 \right), \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t. \quad (2.2)$$

The constraint is more compactly written as $||\beta||_1 \leq t$, and with $\mathbf{y} = (y_1, \ldots, y_n)^T$ we can write (2.2) in a more compact form as

$$\min_{\beta_0,\beta} \left(\frac{1}{2n} ||\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta||_2^2 \right), \text{ subject to } ||\beta||_1 \leq t, \quad (2.3)$$

where $||\cdot||_2$ is the Euclidean norm on vectors, $\mathbf{X}$ an $n \times p$ matrix and $\mathbf{1}$ is a vector of n ones. The predictor matrix $\mathbf{X}$ we often standardize before fitting the lasso, that is with mean $\frac{1}{n}\sum_i x_{ij} = 0$ and variance $\frac{1}{n}\sum_i x_{ij}^2 = 1$. If we would not standardize the lasso solution would depend on the units used to measure the predictors. Though if we would deal with the same unit we would typically not standardize the predictors. The form (2.3) is often rewritten in the so called Lagrangian form

$$\min_{\beta_0,\beta} \left(\frac{1}{2n} ||\mathbf{y} - \beta_0\mathbf{1} - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1 \right).$$

The $\lambda > 0$ can be seen as a shrinkage parameter. By Lagrangian duality there exists a λ that corresponds to the value t chosen in (2.2) that gives the same solution from the Lagrangian form [9, Hastie et al. 2016, ch. 2.2].

Given the logistic model we are interested in estimating the conditional

probability $P(Y = 1|X = X_i) = \mathrm{E}(Y|X = X_i)$, where we turn to maximizing the log-likelihood for a Bernoulli random variable, but which is also equivalent to minimizing the negative log-likelihood with ℓ_1 constraint [9, Hastie et al. 2016, ch. 3.2]

$$
\begin{aligned}
-\frac{1}{n}\sum_{i=1}^{n} y_i \log P(Y = 1|X = X_i) + (1 - y_i)\log P(Y = 0|X = X_i) + \lambda||\beta||_1 \\
= -\frac{1}{n}\sum_{i=1}^{n} y_i(\beta_0 + X_i\beta) - \log\left(1 + e^{\beta_0 + X_i\beta}\right) + \lambda||\beta||_1.
\end{aligned}
$$

2.3 Group lasso

In a situation where some of the predictors have a natural group structure it would be natural to have that either all of the predictors within a group affect the response, or none of them. For example take the case where we have a nominal categorical variable with many levels. It is natural to represent this type of variable of K classes by introducing $K - 1$ new dummy variables where each variable corresponds to one of the different classes [9, Hastie et al. 2016, ch. 4.3]. We would then in the fitting process either want all coefficients for predictors in each group be nonzero or zero. This is when the group lasso would be an ideal candidate. The following is based on [3, Yuan and Lin, 2006] if nothing else stated.

Let us consider a linear regression model with J groups of predictors and denote the explanatory variables for group $j = 1, \ldots, J$ by $X_j \in \mathbb{R}^{p_j}$, p_j is here the size of the group. Then we estimate the response Y with the regression coefficients $\beta_0 \in \mathbb{R}$ and $\beta_1, \ldots, \beta_J$ where $\beta_j \in \mathbb{R}^{p_j}$ by $\mathrm{E}(Y|X)$ which the takes the form

$$
\mathrm{E}(Y|X_1, \ldots, X_J) = \beta_0 + \sum_{j=1}^{J} X_j^T \beta_j.
$$

Then given some sample $\mathbf{y} = (y_1, \ldots, y_n)^T$ and a $n \times p_j$ group matrix $\mathbf{X}_j$ where each row is a realization and each column corresponds to a predictor variable within the group, the optimization problem can be written in Lagrangian form as

$$
\min_{\beta_0, \beta_j} \left(\frac{1}{2}\left|\left|\mathbf{y} - \beta_0\mathbf{1} + \sum_{j=1}^{J} \mathbf{X}_j\beta_j\right|\right|_2^2 + \lambda \sum_{j=1}^{J} ||\beta_j||_{K_j} \right). \quad (2.4)
$$

We thus introduce a new constraint defined as

$$
||\eta||_K = (\eta^T K \eta)^{1/2},
$$

where K is a symmetric $d \times d$ matrix and $\eta \in \mathbb{R}^d$. There are many choices of the matrix K_j, but one proposed in [3, Yuan and Lin 2006] is $K_j = p_j I_{p_j}$. With this choice, we can rewrite the above Lagrangian form

$$\min_{\beta_0, \beta_j} \left(\frac{1}{2} \left|\left| \mathbf{y} - \beta_0 \mathbf{1} + \sum_{j=1}^{J} \mathbf{X}_j \beta_j \right|\right|_2^2 + \lambda \sum_{j=1}^{J} \sqrt{p_j} ||\beta_j||_2 \right).$$

So we get a group penalization depending on the group size. Note that if all groups are of size 1, that is $p_1 = \cdots = p_J = 1$, and setting $K = 1$ gives $||\beta_j||_2 = ||\beta_j||_1$. The regular lasso is then obtained in (2.4). In the setting Yuan and Lin (2006) propose the group lasso, the group matrices $\mathbf{X}_j$ are orthonormal (i.e. $\mathbf{X}_j^T \mathbf{X}_j = I_{p_j}$). Though for general matrices we can orthonormalize them before conducting the group lasso, but this will not generally solve the original problem formulation [5, Friedman, et al. 2010]. Fortunately [6, Yang and Zou 2015] proposed the groupwise-majorization-descent (GMD) algorithm to solve general group lasso problems which depends on that the loss function satisfies the quadratic majorization (QM) condition. The logistic loss is one type of loss function that satisfies the QM condition [6, Yang and Zou, 2015]. It should also be noted that with special data structure the estimated group coefficients in the group lasso may not all be zero or nonzero.

Let us introduce the group lasso for modeling binary responses when using logistic loss. We model the conditional probability $P(Y = 1|X = X_i)$ by (2.1) and the grouped lasso solution is given by minimizing

$$-\frac{1}{n} \sum_{i=1}^{n} y_i (\beta_0 + \sum_{j=1}^{J} X_{ij}^T \beta_j) - \log\left(1 + e^{\beta_0 + \sum_{j=1}^{n} X_{ij}^T \beta_j}\right) + \lambda \sum_{j=1}^{J} \sqrt{p_j} ||\beta_j||_2.$$

The $X_{ij} \in \mathbb{R}^{p_j}$ is a vector of covariates for realization i and group j [4, Meier et al. 2008]. The GMD algorithm uses $-1/1$ to code the class labels so the loss function needs to be reconsidered and thus we introduce the margin based loss. The negative log-likelihood then becomes

$$-\frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i(\beta_0 + \sum_{j=1}^{n} X_{ij}^T \beta_j)}\right) + \lambda \sum_{j=1}^{J} \sqrt{p_j} ||\beta_j||_2.$$

The product $y_i(\beta_0 + \sum_{j=1}^{n} X_{ij}^T \beta_j)$ in the loss is called the margin. A positive margin means a correct classification while a negative margin means an incorrect classification.

2.4 Classification and Regression Trees

Classification and regression trees (CART) is a method where we sequentially use binary splits on the predictor variables X, so that the feature space is partitioned into different regions to be able to predict the response Y. For example suppose we have a regression problem with the two predictor variables X_1, X_2 and want to predict the response Y, here the response is estimated by the mean of Y in each region. Following the right subfigure in Figure 1 we first split X_1 at the value t_1 which results in getting two regions of the feature space. Values of $X_1 \leq t_1$ is sent to the left while values of $X_1 > t_1$ is sent to the right. The first variable and split-point is chosen with respect to some measure of fit, this we will come back to shortly. The same idea is applied in the next step where we split the two regions once more, with splitting-points $X_2 = t_2$ and $X_1 = t_3$. This goes on as long as some stopping criteria is not met. In our example the regions $R_1, \ldots, R_5$ we end up with is illustrated in the left subfigure in Figure 1. Then our prediction of Y would become a constant c_m in region R_m, that is

$$Y = \sum_{m=1}^{5} c_m I((X_1, X_2) \in R_m),$$

where I is the indicator function. The tree model can obviously be used with more than two predictor variables, but for illustrative purposes it is quite difficult to draw the partition of the feature space for more than 2 variables. The following and former description of CART is based on [10, Hastie et al. 2017, ch. 9.2] if nothing else is stated.

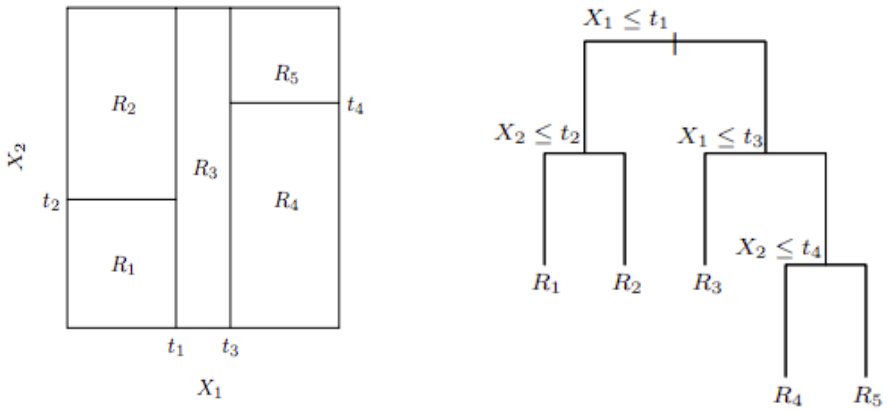

Figure 1: The left figure is an example of how a partitioned feature space using binary splitting may look like. The right figure shows the tree corresponding to the partitioned feature space in the left figure.

Let us now turn our attention to how we build a tree. The approach for constructing a classification tree or a regression tree does not differ much. The only difference is the function used for measuring the splitting fit. So first we introduce the regression tree algorithm and then for the classification tree only change the splitting measure to be more fitting.

Starting with some preliminary definitions, a terminal node is the name of a region we end up with. So in the previous example we would have 5 terminal nodes representing regions $R_1, \ldots, R_5$. A node is a split-point, like the first split $X_1 = t_1$ would be considered a node (more specifically the root node). Now consider a sample $x_i = (x_{i1}, \ldots, x_{ip})$ of size n and the corresponding response y_i. Then we need to decide which variable to split, the split points and also the shape or structure of the tree. For this end suppose we have a partition of the feature space into M regions $R_1, \ldots, R_M$, then a predicted response becomes a constant c_m depending on which region it ends up in. That is we estimate y_i by

$$\hat{y}_i = \sum_{m=1}^{M} c_m I(x_i \in R_m).$$

Then use the sum of squares as a minimization criteria

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

and with this the best constant $\hat{c}_m$ is the mean of all the y_i in the region R_m which is $\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$. The next step is now finding the binary partition with respect to minimizing the sum of squares. This approach is very computationally demanding and hence we need to proceed in a different way. Using all the data we consider splitting variable j at point s so that we get the two half-planes

$$R_1(j, s) = \{X | X_j \leq s\}, \; R_2(j, s) = \{X | X_j > s\}.$$

For the variable j and the splitting point s we are then interested in minimizing

$$\min_{j,s} \left(\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_1 - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right).$$

As before the inner minimizations are solved by $\hat{c}_1 = \text{ave}(y_i | x_i \in R_1(j, s))$ and $\hat{c}_2 = \text{ave}(y_i | x_i \in R_2(j, s))$. It turns out that finding the splitting point s for a splitting variable is quick and hence we can find the best pair (j, s) by looking through all inputs.

A natural question now is how many splits should we continue doing, i.e how large of a tree should we build? Because we have now found the best variable and corresponding split-point creating two regions. Then we can repeat the process for each new region. Clearly it would not be wise to build a large tree because then we could severely overfit the data, but on the contrary, a most simple tree may not be able to explain the complex structure of the data. A seemingly natural way of dealing with this may be that we only split a node if the sum of squares error decreases by some minimum value. This strategy is not suitable since one split that may seem worthless can generate a good split farther down the tree.

The strategy most appropriate is to grow a large tree and stop splitting when we reach some minimal node size. Then we prune the tree, that is remove non-terminal nodes with respect to some cost criteria. Denote the large tree grown by T_0 and let T be some (proper) subtree of T_0, that can be obtained by pruning T_0. Also let m denote terminal nodes, R_m be the region defined by that terminal node and $|T|$ the number of terminal nodes in tree T. Define also $N_m =$ Number of $x_i \in R_m$. Now comes the part of defining the so called node impurity measure $Q_m(T)$, which as the name implies, is a measure of how good the splitting node is. For regression it would be natural to use the squared error loss, but this is not a good choice for a classification task for say $1, \ldots, K$ outcomes. Let $k = 1, \ldots, K$ and define

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

which is the proportion of class k observations in the terminal node m. To assign the predicted outcome to a corresponding class we will use the majority vote, that is the class with largest proportion in a node becomes the predicted outcome. This we can denote by $k(m) = \arg\max_k \hat{p}_{mk}$, which corresponds to c_m for regression trees. With this we introduce 3 different node impurity measures $Q_m(T)$ that may be valid for classification instead of the squared loss used in the regression trees. These measures are misclassification rate, Gini index and cross-entropy defined by:

Misclassification rate: $$\frac{1}{N_m} \sum_{x_i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)},$$

Gini index: $$\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

Cross-entropy: $$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$

Which one of these measures to chose we will come back to, but now we are ready to define the so called cost complexity criteria $C_\alpha(T)$ as

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|,$$

where $\alpha \geq 0$ is a parameter that one need to set and indicates how much one should penalize larger trees. We can see that large α will penalize the complexity and thus results in a smaller tree, the opposite for small α and $\alpha = 0$ will give T_0. Now we choose some sequence of α and find each $T_\alpha \subseteq T_0$ that minimizes $C_\alpha(T)$. One can show that for each α there exists a unique T_α that minimizes $C_\alpha(T)$. To find T_α we will collapse the internal node that gives the smallest per-node increase in $\sum_m N_m Q_m(T)$. This is conducted until we reach the root tree, that is the tree with a single node. We now end up with a sequence of trees that will contain T_α.

The sequence of α depend on the data set at hand which is why it can not be defined in general. The best α we find by using n-fold cross-validation (often 10-fold), choosing $\hat{\alpha}$ that minimizes the cross-validated loss (any $Q_m(T)$ defined for classification in our case) gives the final tree $T_{\hat{\alpha}}$.

So far we have not specified the type of node impurity measure one should use. The Gini index and cross-entropy are both differentiable and hence more valid in numerical optimization. Also these two measures are more sensitive to changes in node probabilities than the misclassification rate. This makes the Gini index or the cross-entropy more suitable when growing a tree, but when pruning the tree either three of the node impurity measures can be used. Often the misclassification rate is used when pruning.

3 Data and model preparation

Before we construct the models we need to do some explanatory analysis of the data set. The difference in our case with respect to [1, Moro et al. 2014] is as mentioned in the introduction that we only can acquire data over 20 of the variables and over the period 2008 to 2010. Note also that the 20 features we have access to are not all the same as the final 22 features that [1, Moro et al. 2014] ended up with using in their construction of the models. For example they have data over variables that contain information on client-bank relationship, such as if a client has a salary account already in the bank. Also information regarding the agent (the one making contact with clients) like his or her experience or how long the agent has worked. These variables presumably contains quite interesting information that would be useful in building our model. Thus our results are not directly comparable with the results attained by [1, Moro et al. 2014].

Still, we have an interesting data set and another approach to the problem. But before getting into the details of these methods let us discuss the data set summarized in Table 1.

variables		
categorical	levels	numeric
job	11	age
marital	3	campaign
education	7	pdays
default	2	previous
housing	2	duration
loan	2	emp.var.rate
contact	2	cons.price.idx
month	10	cons.conf.idx
day of week	5	euribor3m
poutcome	2	nr.employed
y	2	

Table 1: Table showing the explanatory variables, the response (y) and their types. For categorical types also the levels are seen. The levels are calculated by not including missing data as a group class which is included in the unprocessed data set.

Many of the features are self explanatory but some of them will be given a more detailed description below.

- marital: Marital status (divorced/married/single).
- default: Has credit in default? (yes/no).
- housing: Has housing loan? (yes/no).
- loan: Has personal loan? (yes/no).
- contact: Contact communication type of last contact (cellular/telephone).
- month: Month in in which the call was made, January and February month missing. It is not mentioned why that is.
- day of week: Last contact day of the week (weekdays).
- duration: Last contact duration, in seconds.
- campaign: Number of contacts performed during this campaign and for this client.

- pdays: Number of days that passed by after the client was last contacted from a previous campaign.
- previous: Number of contacts performed before this campaign and for this client.
- poutcome: Outcome of the previous marketing campaign (failure/success).
- emp.var.rate: Employment variation rate, with a quarterly frequency compared over same period previous year.
- cons.price.idx: Consumer price index - monthly indicator.
- cons.conf.idx: Consumer confidence index - monthly indicator.
- euribor3m: Euribor 3 month rate - daily indicator.
- nr.employed: Number of employees - quarterly average.
- y: Has the client subscribed a term deposit? (binary: 'yes','no').

The numeric features "emp.var.rate" all the way to "nr.employed" are social and economic context features. For example the "cons.price.idx" corresponds to the consumer price index which is a monthly measure of the average price trend regarding the entire private domestic consumption. The "duration" is the duration of the last call with the client which discussed later, is a problematic feature to use in our models.

For the categorical variables we can see that we are dealing with quite a lot of them and they have many levels (number of possible classes). Note that the "y" is the outcome of the campaign which is the variable of interest to predict. One problem that arises is that we need to decide what type of variables these are (nominal, ordinal) and how to encode them for use in our models.

Before we construct any of the models we need to do some explanatory analysis of the data. We can first of mention that the data set is quite big with 41188 instances, though very imbalanced with respect to the response y. There are some different approaches to solve this problem, but we will make use of random under-sampling. Random under-sampling aims to balance out the data set by randomly removing data from the majority class. One obvious problem with this method is that we may remove data that are very important. Furthermore we can not say that the under-sampled data reflects a random sample from the true distribution, since we have removed data from the majority class in the full data set [11, Kotsiantis et al. 2006]. Another approach would be to use over-sampling based methods where we fabricate synthetic samples to the minority class to balance the data set.

Though for convenience we will use under-sampling since it is simple to implement.

Let us take a more detailed look at the features. Firstly, we note that the duration variable is not desired for use in a predictive model, why we will remove it. The duration of the call impacts the response heavily and is not known before the call is made. Secondly the default attribute is also removed since it only has 3 realizations of people with credit in default. This huge difference in group size will not contribute to the predicative power of the model. We will not do any further feature selection since the models that are used will serve that purpose.

The way of dealing with missing inputs needs to be discussed. We will remove rows with missing values. Even though classification trees can work with missing values the grouped lasso can not. When the rows with missing inputs are removed (roughly 3000 instance) we end up with approximately 38000 realizations out of the full data set. Hence, in the end we are not penalized much when removing the realizations with missing inputs.

Categorical variables are encoded differently for the classification tree and the group lasso. Though binary variables are transformed to 0 and 1 outcomes for both models. The same for the education variable since it is classified as an ordinal type variable why we will use simple ordinal encoding (values 0 to 6). Variables like marital, pdays, day of the week and month are classified as nominal variables and we will introduce $K-1$ dummy variables where K is the number of classes in the variable. For example, day of week will be represented by 4 dummy variables where if all equal 0 this represent that it was a Friday. This is where the difference occurs. The group lasso will have the latter encoding for nominal variables, while a different approach is used for the classification tree.

Regarding the classification tree, if we have a nominal categorical predictor with K classes the number of possible splits are $2^{K-1}-1$. So for large K the computations can become demanding. For our case when we are dealing with a binary response of $0, 1$ outcomes we can instead order the classes as the proportions of falling in the outcome 1. For example, for our job variable we have the class housemaid. Say now that for demonstrative purposes it has 100 observations corresponding to the output 1 and the total number of type 1 outcomes is 500. Then the encoding value of housemaid would become $100/500 = 0.2$. This is done for all the nominal predictors and it can be shown that this gives the optimal split with respect to Gini index [10, Hastie et al. 2017, ch. 9.2.4].

For the group lasso we will make use of the R package called `gglasso` [6,

Yang and Zou 2015]. The default group penalization is set to the square root of the group size, i.e $\sqrt{p_j}$. Standardization of the data is done beforehand. First we will conduct a cross validation with 10 folds to decide the value of λ which minimizes the negative log-likelihood. The sequence of λ values that will be tried are decided automatically by the package `gglasso` but will have an upper bound for λ for which all the coefficient estimates become zero. The smallest λ is decided manually but different values are tried until we get a satisfactory sequence. This will be clear when looking at the cross validated results.

The best model which we will choose is decided by using the one-standard error rule which we for convenience denote 1-SE rule. We pick the simplest model such that the cross validated error is within one standard error above the error of the model which attains the minimal loss. The reason behind the one-standard error rule is to take a conservative approach in the model selection [9, Hastie et al. 2016, ch. 2.3].

The classification tree as mentioned also uses cross validation to estimate the complexity parameter α. The node impurity measure we will use is the Gini index, both for growing the tree and to guide the cost complexity criteria. We will follow the theory discussed in section 2.4 so that we first grow a large tree and then prune it. The one-standard error rule is also applied here to choose a more simple model. Note that for the tree we will make use of the `rpart` [12, Therneau and Atkinson 2022] package in R. For trees it should be noted that no standardization of the input data is needed. This is due to the fact of the binary splits. The tree algorithm tries to find the best way to split realizations, so only the placement of the realizations with respect to each other is interesting. So any monotonic transformation of the data does not change the ordering of the realizations, thus standardization will not be meaningful. The data set is also split into a training set (80 % of data) and a test set (20 % of data).

4 Results

4.1 Group lasso model

Starting with the group lasso, we need to find the best λ so that we minimize the negative log-likelihood. In Figure 2 we can see a 10-fold cross validation for a sequence of λ. The leftmost dotted line indicates the λ which attains the smallest mean error, while the rightmost dotted line is the λ corresponding to 1-SE rule .

It should also be mentioned that the bars indicate the standard error of the mean for each λ and we can see that a smaller λ results in higher vari-

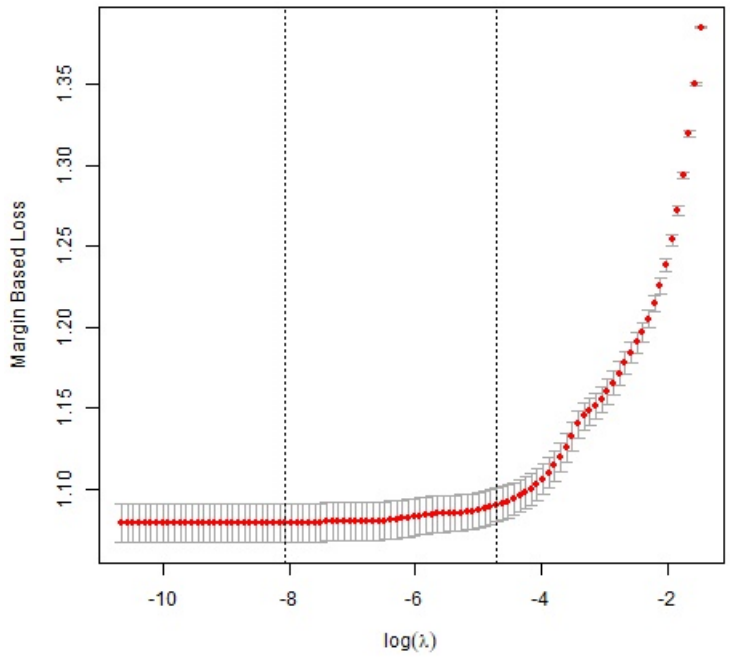

Figure 2: Cross validation error curve for the group lasso. The value of the penalization λ on log-scale is given on the x-axis and the corresponding loss on the y-axis. The λ which gives the minimal loss and the 1-SE rule is the leftmost dotted line and the rightmost dotted line respectively.

ance, while a larger λ results in a smaller variance but greater loss. This is due to the fact that we in general get a more complex model if λ is small. By complex it is meant that the model includes many explanatory variables. So the model overfits the training data and is then quite unstable when it comes to estimating the validation data in the cross validation. For a large λ a simple model will be trained so that often the same and very few variables are included in the model, resulting in low variability in the models trained during cross validation hence a small variance. But often poor predicative power seen clearly in Figure 2. The above argumentation is brought up in favor of the 1-SE rule. We would not like a too complicated model but on the other hand not a too simple model as well. The perfect model is the one in between them in some sense and as a consequence the 1-SE rule is applied here.

The value of λ also directly affects the interpretability of the model. As just choosing the model that minimizes the loss (call this Model 1) will give coefficient estimates that are not interpretable since the model adapts to

noise [9, Hastie et al. 2016, ch. 2.3]. To strengthen this belief we will take a look at the coefficient paths for the sequence of λ seen in Figure 3. The λ which minimizes the loss is the leftmost dotted line while the rightmost is the λ for the 1-SE rule in Figure 3. For the minimal loss all of the coefficients are nonzero. This is a result of not penalizing the coefficients enough and we should hence be careful to interpret any of the coefficients for this λ.

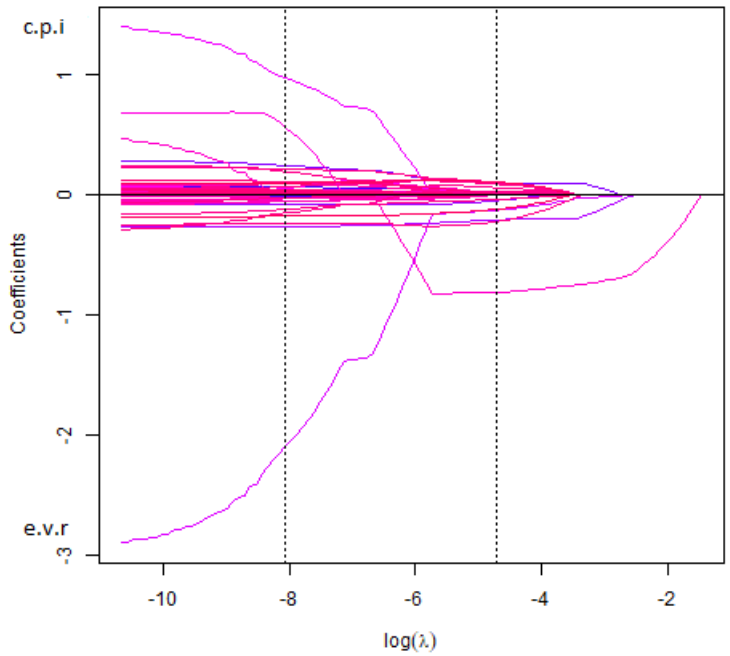

Figure 3: The path of the coefficients of the group lasso model. The value of a coefficient is given on the y-axis as a function of the penalization factor λ given on the log-scale. The λ which gives the minimal loss and the 1-SE rule is the leftmost dotted line and the rightmost dotted line respectively. Note the text on the y-axis corresponding to the coefficients which is the c.p.i denoting consumer price index (cons.price.idx) and e.v.r denoting employment variation rate (emp.var.rate).

Also it could be the case that highly correlated variables are included in the model. This may be present because we can see that some variables have very large estimates (in the sense of absolute value) in Figure 3. Although we can not say for sure if there are correlated variables present this should still serve as a probable cause to use the 1-SE rule, attaining sparser estimates. So we will use the 1-SE rule to choose a more sparse model but not increase to much in training error (call this the Model 2). Both Model 1 and Model 2 will be compared to show that we do not increase that much in error. To be able to compare the models we first need to set the threshold π_0 for the

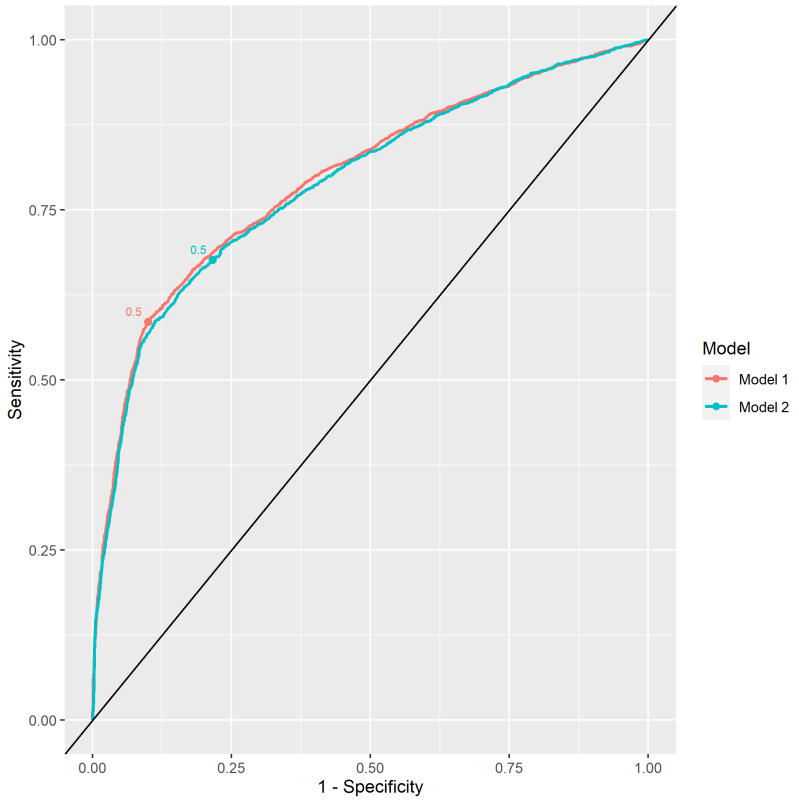

Figure 4: The ROC curves for Model 1 and Model 2 regarding the group lasso. As mentioned before Model 1 corresponds to the model that minimizes the loss, while Model 2 is the 1-SE model. The marked points represents the value of the threshold π_0.

classification rule. A quite natural threshold would be to choose $\pi_0 = 0.5$ so that predictions that are greater than 0.5 are classified as type 1 response, and less than 0.5 is type 0 response. This is although naive since then the misclassification rate depend on what value of π_0 that is set and we surely can not know the best threshold for our model beforehand. To overcome this problem we will introduce a receiver operating characteristic (ROC) curve which plots the sensitivity as a function of $1-$specificity for the possible π_0. But first we need to clarify the terms:

- Sensitivity: probability of predicting 1 given that the true value is 1.
- Specificity: probability of predicting 0 given that the true value is 0.

This would mean if we set π_0 very small say equal to 0, then we would classify all predictions as type 1 response leading to a sensitivity of 1 but a specificity of 0. Obviously, then if we set $\pi_0 = 1$ then we would with similar argument get a sensitivity of 0 but a specificity of 1. The ROC curves for

Model 1 and Model 2 are presented in Figure 4. The diagonal black line corresponds to the ROC curve for a random classifier, essentially meaning random guessing. From Figure 4 we can see that both models have very similar ROC curves which implies that they perform equally well. Since the simpler model (Model 2) has less nonzero coefficients we choose this as our main model, but also because it is the model attained after the 1-SE rule.

In Table 2 we can see all the nonzero coefficient estimates for Model 2. We can see that we end up with many variables still after using the 1-SE rule. One should be careful when interpreting the coefficients because we can at this point see a problem with the group lasso. To my knowledge there are no papers that derive the standard error estimates for the coefficients in the group lasso. Without the standard error we can not really tell which variable is the most important or what type of effect it has on the response. For example, the nr.employed variable seems to have the biggest impact on the response and it has a negative impact. Note that the nr.employed is the total number of employed citizens in the country. But it may be that the standard error is quite large so that the nr.employed is not really an important variable. It could rather be that a seemingly unimportant variable like campaign might be very important if the standard error is relatively small.

intercept	0.09971582	contact	0.1008462
campaign	−0.04974841	pdays	−0.2086955
emp.var.rate	−0.1301319	cons.conf.idx	0.01193242
nr.employed	−0.8130947	housemaid	−0.0003779426
services	−0.0009727713	admin.	0.0003028825
technician	−0.00007787145	blue.collar	−0.001309253
retired	0.001821364	management	−0.0001560224
unemployed	0.0007623251	self.employed	0.0003387860
entrepreneur	0.00004376793	married	−0.003796618
single	0.006662071	May	−0.2232285
Jun	0.05824236	Jul	0.09732272
Aug	0.03170347	Oct	0.08908880
Nov	−0.04007849	Mar	0.1018037
Apr	0.01272875	Sep	0.01294252
failure	−0.1208475	success	0.1053626
education	0.02923792		

Table 2: The nonzero coefficients for Model 2.

In regular circumstances we may expect that the higher number of employed citizens, the more prone people are to subscribe a bank term deposit. If we trust our model it displays the contrary. Note that our data set is over

the financial crisis period and this could be an explanation to this strange behavior, as new research implies which will be discussed later in more detail. It is not only the effect of the number of employed that we find odd. The employment variation rate (emp.var.rate) also seems to have a negative impact on the response. This is also quite peculiar since a positive variation rate displays that the total employment has risen compared to the same quarter previous year. Which we would expect have a positive effect on the response, i.e. it is more likely to subscribe a bank deposit. This could also be argued to be a consequence of the financial crisis or problem with correlation. In conclusion we should be careful when interpreting the results of the group lasso.

4.2 Classification tree model

For the classification tree we as mentioned grow a large tree and then prune it according to the cost complexity criteria discussed in section 2.4. For some sequence of α each corresponding optimal tree is derived and thus its training error. We can plot the cross validated error of a tree as a function of the complexity parameter (cp which is a transformation of α), the tree size (number of terminal nodes) is also shown in Figure 5.

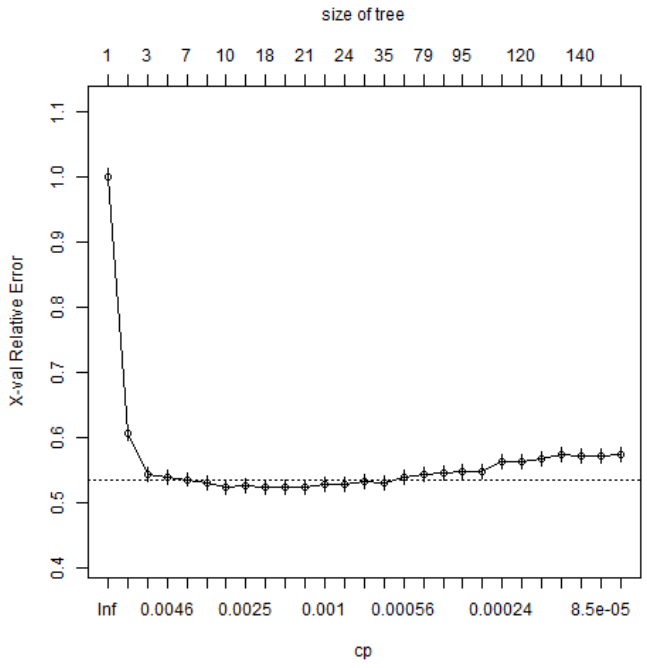

Figure 5: Cross validation (10-fold) error as a function of complexity parameter cp with standard error bars. The dotted horizontal line represents the 1-SE rule. The size of the tree refer to the number of terminal nodes in the pruned tree, that is $|T_{cp}|$, where cp is a transformation of α.

It should be noted that we do not measure the cross validation error in misclassification rate, rather in cross validated relative error. This measure of error does not influence our model selection in any other way than it would if we would have used the misclassification error. This scale is what is used in the `rpart` package to produce the plot. The cross validated relative error is scaled so that the error for the root tree, cp $\in [1, \infty)$, is 1. The cost complexity parameter cp is not the same as α, but is just a transformation. The transformation is used because it has a nice interpretation for regression trees.

Note that the trees in Figure 5 make use of the majority vote to assign a class label to the predicted probabilities. Though we can easily plot the ROC curve for the tree we choose which we will do later. The tree we will choose here is according to the 1-SE rule which is displayed by the dotted line. We can in Figure 5 clearly see that a more complex tree (greater size) does in general not generalize well since it overfits the training data. Another thing to mention is that the cp values are actually ranges for which the same minimizing tree T_α is obtained which is why the leftmost cp is equal to infinity. Meaning the interval $(\alpha_{m-1}, \infty]$ where m is the number of intervals, see [12, Therneau and Atkinson 2022].

Now we are ready to show the structure of our tree. We choose according to the 1-SE rule the tree which we get if we prune it by setting the cp parameter to 0.003247712. The structure of the tree can be seen in Figure 6 which gives that our tree is of size 9. We can see some similarities with the group lasso regarding what variables the tree chooses. Variable importance can be explained in a more robust way for trees, but we can from the structure gain a lot of interesting information. But first we should explain Figure 6. The first number in each node is the class label that results from the majority vote. Under it is the proportion of type 1 labels in the node and under that is the percent of the total data considered in that node. For example we can see in the root node that the majority vote classifies all data as a type 1 response. The proportion of type 1 response is 0.5 and 100% of the data set is considered here.

We can see that the first split is made on the number of employed variable which implies that this variable best explains the response, which seem to coincide with the group lasso. Also note that the campaign and month is also in the tree. But aside from that the tree uses different variables. The tree uses 3 of the social and economic attributes consumer price index, consumer confidence index and euribor3m. It is interesting that the tree uses a lot of these types of attributes in contrast to the group lasso. To summarize we could conclude that the group lasso and tree are quite different. Though just because some of the variables are not included in the final tree does not

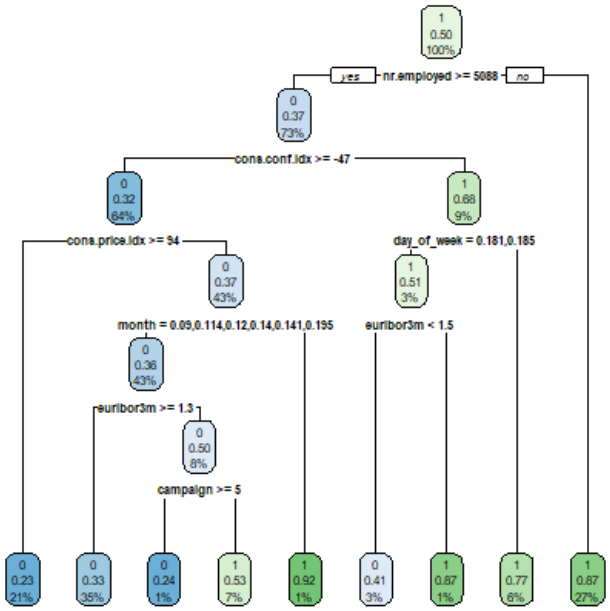

Figure 6: The pruned tree for the optimal cp. First number in each node is the majority vote outcome, second number is the proportion of class 1 responses in the node and the third value is the percent of data in the node at this point.

necessarily mean that they are unimportant. It could be that for example the employment variation rate competes in many of the splits as one of the most important variables but does not give the best improvement and hence is not chosen as a splitting variable. This is when we can look at the variable importance for a tree.

nr.employed	23	euribor3m	22
cons.conf.idx	18	emp.var.rate	13
cons.price.idx	12	pdays	8
month	2	contact	1
day of week	1		

Table 3: Variable importance for the classification tree scaled to add up to 100.

The variable importance is the sum of goodness of split measures in each split where it is considered. In our case this amounts to the improvement of Gini index for a variable considered in a split. The value of the improvement is not that important but rather the relative improvement compared to the

other variables is more meaningful [12, Therneau and Atkinson 2022]. The variable importance can be seen in Table 3 rounded and scaled so it adds up to 100. Variables with a proportion of less than 1% are omitted. We can see that the nr.employed has the biggest importance but closely followed by euribor3m. The variable importance show that variables that are not included in the final tree still play an important role. The emp.var.rate is not included in the final tree but has a high variable importance. So both the tree and group lasso seems to agree more than previously thought. The pdays is included in the group lasso but not in the tree but it has a relatively large variable importance. Both of these variables have relatively large estimates in the group lasso which may indicate that the models somewhat agree on important variables.

4.3 Model assessment

In previous chapters we have discussed the results of the two different models. Now we would be interested in their predicative power and which of the models we should choose. The first step is to find the optimal threshold value π_0 for our problem. What value gives the best trade off between sensitivity, specificity and prediction accuracy? The prediction accuracy is the proportion of correct classifications on the test data set. Taking into account that the data is over the financial crisis period the bank would be interested in creating successful sales of long-term deposits even if it means spending more time contacting non-buyers. Hence we would put more focus on obtaining a high sensitivity rather than specificity.

Figure 7 displays the ROC curves for the classification tree and the group lasso model on the training data. The threshold $\pi_0 = 0.5$ is displayed but note that it is rounded to the closest point that makes the same class label assignments. From this we can immediately notice a drawback for our classification tree. The problem with having a simple tree is that we do not have an abundance of thresholds that are valid to set. This is due to the fact that we predict the label with the majority vote in the terminal node the observation end up in. So we can at most have 9 possible threshold values that give different results, the same amount as the number of terminal nodes in our tree (Figure 6). That is why the ROC curve for the tree has the piece-wise linear behavior. The group lasso however does not have the same problem.

The different choices of threshold for the tree model are not that many. To get any higher sensitivity than setting $\pi_0 = 0.5$ the next choice is $\pi_0 = 0.33$, which is quite a big step. In Table 4 we can see the confusion matrix for the tree and the group lasso for the two different thresholds on the test data set. With the confusion matrix we can calculate the specificity, sen-

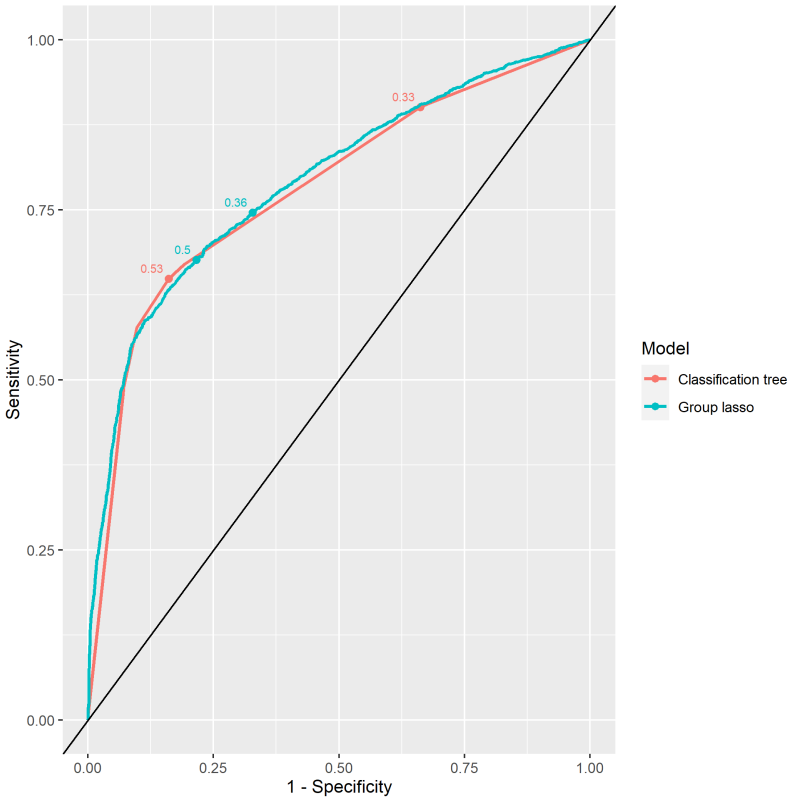

Figure 7: The ROC curves for the classification tree and the group lasso model on the training data. Marked points on the lines correspond to the threshold value π_0.

sitivity and prediction accuracy. Starting with the tree with $\pi_0 = 0.5$, the sensitivity is equal to approximately $520/833 \approx 0.62$, the specificity $712/870 \approx 0.82$ and prediction accuracy 0.72. The tree with $\pi_0 = 0.33$ has sensitivity $752/833 \approx 0.9$, specificity $278/870 \approx 0.32$ and accuracy 0.6. We lose a lot of predicative power (12 % units) with $\pi_0 = 0.33$ compared to setting $\pi_0 = 0.5$.

Now looking at the group lasso for $\pi_0 = 0.5$ we can see that sensitivity is $552/833 \approx 0.66$, specificity $680/870 \approx 0.78$ and a prediction accuracy of 0.72. When $\pi_0 = 0.33$ we get sensitivity $683/833 \approx 0.82$, specificity $449/870 \approx 0.52$ and a prediction accuracy of 0.66. We notice that the group lasso for $\pi_0 = 0.5$ has higher sensitivity but lower specificity than the corresponding tree. They have the same accuracy which may make the group lasso be the better choice for our purpose when $\pi_0 = 0.5$. For the other threshold we have the other way around where the sensitivity is higher for the tree than the group lasso. We lose quite a lot specificity for the tree

		True		
		yes	no	Total
Predicted	yes	520	158	678
	no	313	712	1025
	Total	833	870	1703

(a) $\pi_0 = 0.5$ for tree

		True		
		yes	no	Total
	yes	752	592	1344
	no	81	278	359
	Total	833	870	1703

(b) $\pi_0 = 0.33$ for tree

		yes	no	Total
Predicted	yes	552	190	742
	no	281	680	961
	Total	833	870	1703

(c) $\pi_0 = 0.5$ for group lasso

		yes	no	Total
	yes	683	421	1104
	no	150	449	599
	Total	833	870	1703

(d) $\pi_0 = 0.33$ for group lasso

Table 4: Confusion matrices for some different thresholds for the group lasso and classification tree on the test data. Predicted class is the rows for each sub table and the corresponding true class is the columns.

compared to the group lasso and the group lasso does have a bit higher prediction accuracy. The tree however is a much simpler model than the group lasso and has the benefit that we can interpret the model.

5 Conclusion

The group lasso and the classification tree performs similar in terms of the prediction accuracy reaching 72% for $\pi_0 = 0.5$, however the group lasso performs better with respect to other measures. It has a better sensitivity and specificity for our type of problem where sensitive models are desired. Choosing the smaller threshold is not favorable because although we gain a higher sensitivity, the prediction accuracy is heavily affected with the biggest impact on the tree where we lose 12 % of the accuracy (measured in units of %). So for both models the less sensitive model is more beneficial. The prediction accuracy is quite close to the best model (neural network) chosen by [1, Moro et al. 2014] which achieves 75% prediction accuracy on their data set.

We noted that we should be careful when interpreting the group lasso, but the tree could be interpreted which shows that the number of employed citizens is the most important variable and also has a negative impact on the response. It is closely followed by the variable euribor3m as the variable with the second largest variable importance. This variable is not even included in the group lasso. It can also be noted that other variables, e.g. consumer price index (cons.price.idx), are not included in the group lasso but is part of the classification tree. Although the two methods achieve similar results,

they differ quite a lot.

From the tree we can find that all of the variables that are related to economic and social attributes for the country have the largest variable importance. It is hence quite clear that these type of variables can describe the response well. This may also be expected since these measures are developed to explain the economic and social climate in the country and could be quite directly related to the success of selling long-term bank deposits. This is supported by [1, Moro et al., 2014] because 3 of the top 5 variables with highest relative importance (different from our variable importance) are not specific values for individual clients, like Euribor rate (euribor3m) which had the second highest variable importance in this thesis and highest relative importance in [1, Moro et al. 2014]. They use sensitivity analysis on the neural network to measure global influence of an input variable to obtain the relative importance. The other top 2 variables in the sensitivity analysis are not included in our data set. These variables are the call direction (inbound/outbound) and how long the agent has worked in the call center.

The Euribor rate has as mentioned a large variable importance and one would think that a higher rate would increase the savings rate since many European banks align deposit interest rates with the Euribor rate. The classification tree displays the contrary. The splits made in the tree (Figure 6) shows that higher rate is connected to a lower probability of a success. This is although in line with [1, Moro et al. 2014] and can be explained. Research indicates that prior to 2008 there was a positive relation between offered rate for deposits and savings rate. After 2008 when the financial crisis hit this relation reversed so that more bank term deposits where made but the Euribor rate fell. This may be due to that clients feel that saving for the future under a financial crisis is prioritized over spending money in the present crisis. This research could also explain the negative relation between the number of employed citizens and the response since the employment decreases over the time period but the savings rate increases. Prior to 2008 one would probably expect that the number of employed citizens and savings rate have a positive relation just as the relation between offered rate and savings rate.

In conclusion both the tree and group lasso are both valid models for predicting bank marketing success, although the group lasso may be preferred for our problem since more sensitive models are desired. But the gain in sensitivity is not incredible for the group lasso compared to the tree, also we can more easily interpret the tree model while we need to be careful with the group lasso. In a real world environment, a client's reasons for subscribing the bank deposit may alter through time so it would probably

be more desirable to make use of the tree model to be able to detect and validate these changes. For example the economic environment may change over time. Still with an accuracy of 72% the bank can benefit from implementing the tree model to increase efficiency in the call centers by making less unnecessary calls. The group lasso have problems that may have been noticed throughout the thesis.

For the group lasso some problems are present that limits its usefulness as a model used in practice. First of all the group penalty choice is arbitrary for general group matrices ($\mathbf{X}_j$ in section 2.3). There is some research on the group lasso mentioned throughout this report and more research not mentioned here, but to my knowledge none of the reports mention the choice of the group penalty factor for general problems. The original authors [3, Yuan and Lin 2006] to the group lasso recommend to penalize the groups according to the square root of their size, $\sqrt{p_j}$, but under the assumption of the group-wise orthonormal condition. In general this is clearly not the case. An interesting topic, but outside the scope of this thesis, would be to derive theoretical results that would serve as a guide on the value of the group penalization for general data. For our data when we use the recommended penalization, this may give misleading results. The groups may not be penalized as strongly as they should so the interpretation of the group lasso may not be valid. This could also cause some odd interplay between variables that are important and big groups which are not important but seem important since they are not penalized correctly.

Furthermore, there is to my knowledge no papers that propose a method on estimating the standard error for the coefficients in the group lasso. Hence it is difficult to derive the effect that the different coefficients have on the response. That is why more research has to be conducted in this direction.

6 References

[1] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, pp. 267–288, 1996.

[3] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society*, vol. 68, pp. 49–67, 2006.

[4] L. Meier, S. van de Geer, and P. Bühlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society*, vol. 70, pp. 53–71, 2008.

[5] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," 2010.

[6] Y. Yang and H. Zou, "A fast unified algorithm for solving group-lasso penalize learning problems," *Statistics and Computing*, vol. 25, pp. 1129–1141, 2015.

[7] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*. Taylor & Francis, 1984.

[8] R. Sundberg, *Lineära Statistiska Modeller*. Stockholm University, 2020.

[9] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity*, 1st ed., ser. Chapman and Hall/CRC Monographs on Statistics and Applied Probability. CRC press, 2016.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2017.

[11] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, 2006.

[12] T. M. Therneau and E. J. Atkinson, *An Introduction to Recursive Partitioning Using the RPART Routines*, Mayo foundation, 2022.