The Report Committee for Chuanfeng Yang
Certifies that this is the approved version of the following report:

Predicting Success of Bank Telemarketing with Classification Trees and Logistic Regression

APPROVED BY
SUPERVISING COMMITTEE:

Mingyuan Zhou, Supervisor

Kishore Gawande

Predicting Success of Bank Telemarketing with Classification Trees and Logistic Regression

by

Chuanfeng Yang, B.S.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Statistics

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

Dedicated to my parents and Lie Li.

Acknowledgments

First I would like to express my heartfelt thanks to my supervisor Professor Mingyuan Zhou, for his consistent help in my graudate studies. His care and encouragement lead me to pursue my interest in statistics. I would also like to thank Professor Kishore Gawande for his guidance and patience. He taught me learn how to come across difficulties and solve issues step by step. This master report would not have been completed without their help.

I am also grateful for The University of Texas at Austin, where I started graduate studies and began to embrace a new world. Thanks to my friends Yutong Duan, Xinqi You, Lingjia Zhang, Daniel Woodie and Jaclyn Peek. It becomes my beautiful memory the time we spent studying, doing projects and enjoying together.

I would also like to thank my loving parents for their always being there to support and enlighten me. Thanks to my boyfriend Lie Li for his selfless devotion.

Predicting Success of Bank Telemarketing with Classification Trees and Logistic Regression

Chuanfeng Yang, M.S.Stat
The University of Texas at Austin, 2016

Supervisor: Mingyuan Zhou

Success of bank marketing campaign is predicted with customer features, campaign information and economic attributes. To predict whether or not clients will subscribe long-term deposit, logistic regression is applied with backward variable selection and principal components analysis. Random forests and stochastic gradient boosting, as members of classification trees, are also built as comparisons. Based on visualization and quantitative predictive performance, gradient boosting (AUC = 0.791) is slightly better than the other two models. Variable importance from 3 models remains consistent for most variables. Social and economic attributes, such as *euribor3m*, are among top important variables.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In bank industry, forecast plays an important role such as detecting credit fraud to reduce financial loss. Predicting client behavior, on the other hand, will bring profit. Based on features of clients and marketing campaign, a bank need to foresee whether or not clients would subscribe long-term deposits. This can be summarized as a two-class classification problem.

In terms of predicting marketing success, accuracy is obviously important. From marketing perspective, another key issue is that science insights are applicable for business to understand and put into action. The two sides above require both good model predictive performance and model interpretability. Unlike classical statistical models, some data-mining models allow more flexibility and complexity but suffer from lack of interpretability like 'black-box'. In this report, we try to explore two different models (random forests and stochastic gradient boosting) to discuss two aspects of classification models and their comparisons.

This report implements logistic regression as classical statistical model for binary outcome issue. It is popular for its ability to produce probability estimate, which is flexible to make classification by setting threshold. With coefficient estimates, it is also easy to use *odds ratio* to facilitate business decisions directly. Classification trees, as a type of relatively novel method compared to logistic regression, are popular because of ensemble idea to reduce forecast variance and mimic human decision process. Random forests and gradient boosting are brought up as tree-based models to achieve good performance. In addition, variable importance will be explored to complement lack of interpretability. Three models will be compared with quantitative measure (e.g. area under ROC curve) and information about featues will be generated during model fitting for marketing suggestions.

Chapter 2

Literature Review

Moro, Cortez and Rita(2014) propose four data mining models to forecast telemaketing success - logistic regression, decision tree, neural network and support vector machine. The four models are tested against the most recent data with rolling window scheme by area under receiver operating characteristic curve and lift cumulative curve. Neural Network gives the best performance with AUC = 0.8. Support Vector Machine ranks the second best. Logistic regression has the lowest AUC.

Vajiramedhin and Suebsing (2014) suggest data balancing technique for prediction on the same data set. It focuses on proper random sampling for data split so that response labels in training data and test data remains consistent. AUC improves by 4.6% after applying data balancing method.

Prediction for bank telemarketing falls into classification problems and models besides those mentioned above can be explored to either improve the interpretability or potential forecast accuracy. Decision tree becomes popular for its learning ability and more interpretable than 'black-box' methods. However, it suffers from overfit and unstable results. Random Forest and Gradient Boosting are two methods based on decision tree. They both generate smaller but multiple trees to control tree depth to prevent overfit and reduce variance. Variable importance will be extracted to understand ensemble of trees. Logistic regression is also implemented as a classical model for comparison whose interpretability is further discussed.

Chapter 3

Methdology

3.1 Data Pre-processing

Telemarketing dataset is an open dataset in University of Irvine Machine Learning Repository named 'Bank Marketing'. Data are collected from 2008 to 2013 from a Portuguese retail bank with 41188 observations. Features of clients and campaign with additional economic attributes serve as 20 predictor variables.

Response variable is whether or not the client will purchase the long-term deposit, i.e., "yes" or "no". *Duration* represents last contact duration and response will be known at the end of the call which highly influences the response. As primary objective is to predict success rate, this variable should be discarded.

Literally, there is no missing data. Random forests and gradient boosting can deal with missing values but logistic regression cannot. After looking into variables in details, factors *education*, *job*, *marital*, *housing*, *loan* and *default* have 'unknown' level. These also cause categorical predictors to have sparse and unbalanced distribution. Logistic regression is much more sensitivie to this kind of variables than random forests and gradient boosting do.

We would expect logsitic regression performs better after near-zero variation perdictors discarded or near-zero volume levels removed. Hence, data preprocessing is initiated to improve performance.

Default denotes whether the client has credit, which has 3 observations as "yes", 32588 as "no" and 8597 as "unknown". "Unkown" in *default* can reveal some hidden information when clients leave that question blank and it is not as askew as the other five. Thus, "unknwon" level in *default* is kept while "unknown" in other 5 predictors will be treated as missing values. The number of complete cases is 38245 and still reserves 92.9% of the original dataset. Imputation is not necessary in this case as it could also result in uncertainty. Observations with level "unknown" will be removed to ensure data quality. New Dataset with 38245 observations, 19 predictors and 1 response variable will be utilized for model training and test.

To prevent overfitting, we split 38245 observations into training data(80%) and test data(20%). As response variable "yes" only takes a small proportion, data partition by function *createDataPartition* is applied to ensure balanced response in training data and test data. That is to say, random sampling is within each response class to ensure equivalent class distribution.

3.2 Classification Trees

3.2.1 Introduction to Classification Trees

Tree-based models are made up of nested **if-then** statements. Predictors are used to bipartition the data and final data space will be divided into many different-sized rectangular regions. If we look at a tree upside-down from root to leaves, each split is like a branch. It visually shows the approach to recursive binary splitting. It is greedy because best split is made at each step by looking for the predictor giving largest improvement. Tree-based methods can be used for classification as well as regression. We will focus on classification trees since the problem of interest is two-class problem.

Classification models generate a continuous valued prediction between 0 and 1 regarded as the probability of a sample belongs to class one. Then a prediction class is made according to certain threshold which will create a discrete category for predicted response. The threshold highly impacts the result and will be discussed further in classification evaluation.

The purpose of classification trees is partitioning data in a way that places samples into smaller and relatively homogeneous buckets. Homogeneity here,equivalent as purity, means large proportion of samples in terminal node coming from same class. As misclassification mislead us to focus more on accuracy other than purity, there are two major measures to define purity: Gini index(Breiman et al. 1984) and cross-entropy. Gini index for a node is as

$$p_1(1-p_1)+p_2(1-p_2) \tag{3.1}$$

where p_1, p_2 are probability of Class 1 and 2, respectively. Obviously, they sum up to 1. It is minimized at $p_1 = 0$ or $p_1 = 1$, where all samples in the node belongs to one class. When random splitting is made, i.e., $p_1 = p_2 = 0.5$, Gini index is maximized. The aim of trees is to increase purity so best split is choosing predictors that can decrease the index by largest degree.

The tree-growing process described above will stop until number of samples in node falls below certain minimum node size in practical. Once we have a full-grown tree, overfitting comes out as the hurdle to have good performance on test data. There is a trade-off between complexity of tree and goodness of fit on test data. Therefore, pruning depth of tree (complexity) is required. Depth and other parameters will go through tuning process before final predictive model is built.

Classification trees have many advantages. First, if-then conditions like human decision-making are easy to interpret visually and implement. It can also deal with many types of variables regardless of categorical, continuous or askew. In perspective of model specification, it works well even though users do not specify the relationship between response and predictors. On the other hand, it tends to give sub-optimal result if high-dimensional rectangulars cannot delineate the relationship very well. This disadvantage and advantage of not specifying relationship are like two sides of a coin. In addition, it is sensitive to data changes. A slight change in the data can reshape construction of the tree.

To substantially improve predictive performance of classification trees, bagging is introduced as averaging over multiple trees to reduce variance. A general term to describe combination of models is *ensemble*. In brief, bagging generates B boostrap samples to build B decision trees. The majority vote from

B trees acts as the final prediction as it appears more commonly. Bagging fits model based on boostrap samples so observations not used in a given bagged tree are out-of-bag (OOB) samples. These remaining samples will be used to calculate OOB error.

3.2.2 Random Forests

Random Forests provides an improvement on bagging since it decorrelates the trees to a certain degree. As all predictors are taken into account at every split, the bagged trees are not completely independent. Trees, though coming from different boostrap samples, may share similar constructure due to the underlying relationship between response and predictors.This is called tree correlation and will prevent bagging from further improvement. Introducing randomness into tree learning process is proposed and Dietterich (2000) suggested using a random sample of top m predictors from P original predictors. Breiman (2001) proposed to randomly pick k predictors at each split and named the new method with *random forests.* The number of randomly selected predictors is referred to as m_{try}. Commonly m_{try} is set to be approximate square root of the total number of predictors $m_{try} \approx \sqrt{P}$ but evenly spaced values between 2 and P can be tried.

The tweaks on algorithm give improvement in two aspects. As trees are built on boostrap samples and random draws are taken from predictors, tress are more independent and thus reducing variance. Unlike simply selecting the best predicor to improve purity at each partition, many predicotrs can be taken into consideration. That is, almost all predictors have the chance to be the one defining a split. Considered from a local predictors bucket, predictors have fewer competitors and higher chance to get selected. This supports bias reduction.

Compared to bagging, random forests is also welcomed for more efficient computation. Both methods can be implemented in parallel computing. In addition, random forests only needs to make choice from a subset of original predictors for best split. Although it may require more trees for ensemble, it outperforms bagging in computational time.

Algorithm 1 Random Forests algorithm

```
1: Select the number of trees to build, m
2: for i = 1 to m do
3:     Generate a boostrap sample from original data
4:     Train a tree model on this sample
5:     for each split do
6:         Randomly select $m_{try} (< P)$ predictors from originals
7:         Select the best predictor from $m_{try}$ predictors and
8:         Partition the data
9:     end for
10:    Use tree model stoping criteria to determin when a tree is complete
11: end for
```

Random forests has the following properties:

- As member of tree-based models, types of predictors are flexible.
- The model is relatively not very sensitive to m_{try}.

- Out-of-bag measures are available for accuracy, confusion matrix, specificity and sensitivity.

3.2.3 Gradient Boosting

Boosting was originally developed for classification problems starting from AdaBoost algorithm and then evolved to stochastic gradient boosting (Friedman et al., 2000). The idea of boosting came from learning theory where multiple weak classifiers are combined or boosted to produce an ensemble of classifiers. Adaboost, as implementation of learning theory, aims at training (boosting) weak learners into strong learners. Friedman discovered the connection of Adaboost algorithm to statistical ideas, such as logistic regression and loss functions. Boosting is further viewed as an additive model minimizing exponential loss. Then, statistical concepts are generalized to develop a more adaptable method "gradient boosting machines" to solve different kinds of problems.

Given a loss function, gradient boosting picks a weak learner (e.g., classification trees) and develop an additive model to minimize the loss function. Gradient is calculated, which is residual in classification. Model is built to fit the gradient while minimizing loss function. Model follows last model one by one to fit residuals left from last model. All models are then combined to give final output.

First requirement for boosting is a weak learner. As for models with tuning parameters, we can always have a relative strong or weak learner by adjusting parameters. Trees can be a perfect one since its depth highly influence the performance. That is, if we restrict the complexity of trees to be very small, these trees are able to serve as candidate for weak learners. Second, for generating additive model, trees are easy to add and generate prediction. Last but not least, trees are computationally efficient. As a result, trees are perfect weak learners for boosting.

As mentioned above, tree depth is a tuning parameter. Another parameter for simple boosting is interaction depth, which is also known as number of iterations. It is named with "interaction" since predictors used in the subsequent tree can be considered as interaction with all previous predictors.

Gradient boosting has over-fitting drawback since it always tends to pick optimal learner at every step. To prevent this greedy strategy from over-fitting, learning rate is introduced to control. This kind of shrinkage is often denoted by symbol λ. Small learning rate is better to apply but also time-consuming as each step is tiny. Choosing a proper value for λ is also balancing the trade-off between computation time and model performance.

Inspired by boosting algorithm, using boostrap samples not only reduce variation but also saves computation time. Stochastic gradient boosting is evolved by adding random sampling. Naturally, bagging fraction also becomes a tuning parameter. That is, how many percentage of training data should be used at each tree, default at 0.5.

One way to model event probability as stochastic gradient boosting is similar to logistic regression, by

$$\widehat{p_i} = \frac{1}{1 + exp[-f(x)]}, \tag{3.2}$$

where domain of $f(x)$ is $(-\infty, \infty)$. In logistic regression, $f(x)$ is a linear combination of predictor variables. To initialize f_0, we calculate sample proportion of class p for sample log odds $log\frac{\widehat{p}}{1-\widehat{p}}$.

In bank telemarketing problem, Bernoulli distribution is appropriate to describe two-class probability. The algorithm for stochastic gradient boosting is shown below.

Algorithm 2 Stochastic Gradient Boosting algorithm for 2-class classification

```
1: Initialize all predictions to sample log-odds as $f_i^{(0)} = log\frac{\widehat{p}}{1-\widehat{p}}$
2: for  iteration $j = 1$ to $M$ do
3:     Compute gradient (i.e. residual) $z_i = y_i - \widehat{p_i}$
4:     Randomly sample training data
5:     Train a tree on the random sample with residuals as outcomes
6:     Compute the terminal node estimates of Pearson residuals:
7:     $r_i = \frac{1/n \sum_i^n (y_i - \widehat{p_i})}{1/n \sum_i^n \widehat{p_i}(1-\widehat{p_i})}$
8:     Update the current model using $f_i = f_i + \lambda f_i^{(j)}$
9: end for
```

Summarized from the algorithm, we need to tune 3 parameters. Thus, a tuning parameter grid is constructed as combinations of different number of trees M, interaction depth and shrinkage rate λ. We will pick optimal combination to fit gradient boosting for training data.

Comparing algorithm of Random Forests to Stochastic Gradient Boosting, there are two similarities. Trees are used as base learners though weak learners for boosting. Final result is based on multiple trees (i.e. ensemble). However, significant differences make them two separate methods. In random forests, trees are independently generated and created to be fully grown. All trees will be given same weight in account of final prediction. In boosting, trees are dependent on past ones and supposed to have small depth as weak learners. Earliy trees will receive more consideration for final prediciton. Another difference coming from independence of trees in random forests is that trees can be built in parallel and less time required.

3.3 Logistic Regression

3.3.1 Introduction to Logistic Regression

To estimate probability of event, subscription of deposit in this situation, sample proportion is the maximum likelihood estimate (MLE). Subscription is also affected by many other factors such as customer occupation. Thus, event probability p is a function of those factors. Based on linear regression, transformation of p can be described as a linear combination of factors. This is the general idea behind logistic regression. Similarly, it has an intercept as well as slope parameters for each term. In common linear regreesion, response variable y can take value on $(-\infty, \infty)$ whereas probability is within (0,1). To restrain fitted probability within reasonable range, log odds of the event rate replaces the common y for response.

We first introduce odds as ratio of probability of event rate over probability of nonevent rate. To quantify predictive ability of a predictor, odds ratio is a common method. Suppose we have odds of Monday in day of week as γ_1 and odds of other non-Mondays in day of week as γ_2. The ratio of the odds $\frac{\gamma_1}{\gamma_2}$ denotes that odds of subscription when last contact on Monday is $\frac{\gamma_1}{\gamma_2}$ times larger than odds of non-Mondays. Odds ratio of Monday over non-Mondays is $\frac{\gamma_1}{\gamma_2}$. Given high odds ratios, we would expect predictors to have larger impact on classification.

Recall event probability p, odds of p is $p/(1-p)$. Log odds of p regresses on predictors:

$$log\frac{p}{1-p} = \beta_0 + \beta_1 x_1 + ... + \beta_P x_P. \tag{3.3}$$

Here, $P = 19$ is number of predictors. By transforming back to p, we have:

$$p = \frac{1}{\beta_0 + \beta_1 x_1 + ... + \beta_P x_P}.$$

Plot of event probability is a sigmoidal function and p is constrained within (0,1). When factor levels are not significant, we will combine some factor levels to increase model fitting. For numerical variables, we also categorize them to make more sense for significance. Hosmer and Lemeshow Goodness of Fit test is used to test model fit. Original hypothesis is model fits the data very well and we would not reject the original hypothesis if p-value > 0.05.

3.3.2 Backward Variable Selection

When some levels in a categorical predictor is not significant, we still keep the predictor. Backward variable selection is a method to remove numerical variables or categorical variables that none of the levels is significant.

Backward selection starts with the full set of predictors and sequentially deletes the smallest-impact predictor in the model. We drop the candidate variable with smallest Z-score. If all predictors in the model have significant Z-score (i.e. p-value < 0.05), backward selection stops. Note that backward selection is applicable only when number of observations is larger than number of predictors. Function *step* in R is used with option "backward".

3.3.3 Principal Components Analysis

Sometimes variables have strong correlation (i.e. multicollinearity) and it gives rise to high variance of estimates. Principal components analysis (PCA) produces linear combinations of variables in sequence. These principal components have maximal variance and are mutually independent. In a hyperplane, vectors in represent of the principal components are perpendicular to each other.

To obtain principal components, center variables to zero and scale them to have unit variance. Hence, we will not put more loading on a variable simply becuase of its large volume. By solving singular value decomposition of $\mathbf{X}(x_1, ..., x_q)$, we obtain q principal components.

For first principal component Z_1 of a set of features, it is the normalized linear combination of features which has largest variance. By normalized, it means that loadings on all features sum up to 1. Second principal component is the one has second largest variance and independent of first principal component. First principal component:

$$Z_1 = \Phi_{11}X_1 + \Phi_{21}X_2 + ... + \Phi_{q1}X_q$$

where $\sum_{j=1}^{2} \Phi_{j1}^2 = 1$. We select top q^* principal components and replace original set of features with them. To determine how many components to choose, proportion variance explained by chosen ones usually add up to 80% or above.

3.4 Model Interpretation and Predictive Performance Evaluation

3.4.1 Variable Importance

For classification trees, variable importance can be extracted by 2 different measures. The general idea is that for each single tree and each predictor, improvement in the predictive objective because of that predictor is aggregated. Then improvements are aggregated over trees for that predictor in ensemble to give an overall variable importance.

One method is obtained by permuting out-of-bag data. Prediction error on OOB data is stored for each tree. Then permute each predictor and calculate the same thing. The difference between the unpermuted errors and permuted ones will be averaged over all trees and divided by standard variation of the differences.

Another measure is using Gini index to calculate total purity increase from splitting, averaged over all trees. Recall impurity measure Gini index $p_1(1-p_1)+p_2(1-p_2)$. In each split, samples are sorted by predictor variables and split will be made at the midpoints of predictors. The optimal split is try to decrease Gini index as much as possible. As for binary response, a two-way contigency table is represented for a split in Table 3.1.

Prior to the split,

$$Gini(\text{prior to split}) = 2(\frac{n_{1+}}{n})(\frac{n_{2+}}{n}).$$

After the split, Gini index is first generated in each new node and then weighted

Table 3.1: Split illustration

	Class 1	Class 2	
branch 1	n_{11}	n_{12}	n_{+1}
branch 2	n_{21}	n_{22}	n_{+2}
	n_{1+}	n_{2+}	n

by sample proportion. Simplified Gini(after split) would be:

$$Gini(\text{after split}) = 2[(\frac{n_{11}}{n})(\frac{n_{12}}{n_{+1}}) + (\frac{n_{21}}{n})(\frac{n_{22}}{n_{+2}})].$$

In random forests, within each tree ($Gini$(prior to split) - $Gini$(after split)) is aggregated for each predictor based on split criteria. Then it is aggregated over all trees in the forest. The larger the sum, the more important the predictor variable is. As for boosting, it works in the similar manner.

Tree-based models, especially ensemble of trees, usually suffer from lack of interpretability. Variable importance will reveal more visibility about the model and provide insights for business. We will compare variable importance by two different measures to see how they align. In addition, variable importance from random forests and gradient boosting will also be examined together. To give it further evaluation, variable significance given by logistic regression can also give a hint about impact of predictors. In logistic regression, Z statitic and p-value are indicators of significance. Variables making a difference in classification trees and logistic regression may reveal interesting insights.

3.4.2 Predictive Performance Evaluation

When implementing diversified models, approach to measure predicitive performance is important for model comparisons. Logistic regression model is favored because of simplicity. Classification trees are favored since they allow more flexibility about model hypothesis and better performance. Due to different procedures, (e.g. variable selection in backward logistic and boostrap samples from random forests) finding unbiased way to evaluate models and suggest preferred models will be discussed in the following ways.

3.4.2.1 Confusion Matrix

For a two-class problem, the outcome can be summarized as "event" and "nonevent", where event refers to result of interest. Event in this case is client will buy the long-term deposit in bank marketing problem. A common method is confusion matrix. A 2×2 cross-tabulation is a good choice for decribing performance. Values on the diagonal means number of cases that have been correctly classified, whereas off-diagonal values denotes number of those not.

Table 3.2: Confusion matrix

Predicted	Observed	
	Event	Nonevent
Event	TP	FP
Nonevent	FN	TN

True Positive(TP) = samples with event and predicted to have the event

False Positive(TP) = samples without event and predicted to have the event

3.4.2.2 Accuracy Rate and Kappa Statistic

An intuitive method based on confusion matrix would be overall accuracy rate. It indicates by how much predictions agree with observations. With definitions from table, overall accuracy rate is $\frac{TP+TN}{TP+FP+FN+TN}$. This is straightforward but would bias us to weight different types of errors equally. In bank telemarketing, the cost of erroneously not contacting a highly possible client is likely to be higher than incorrectly contacting a nearly impossible client. Moreover, we need no-information rate (e.g. 50% for random guess) as baseline for measurement. For example, if response "yes" has very low frequency, it is safe to predict all samples to have response "no" with a small enough accuracy rate. One type of no-information rate is the percentage of most majority class in training data.

Combining accuracy rate with no-information baseline, Kappa statistic is developed as

$$Kappa = \frac{O - E}{1 - E}$$

where O is observed accuracy and E is expected accuracy. Value of Kappa statistic range from -1 to 1; $Kappa = 0$ denotes no agreement between observations and predictions. The larger Kappa is, the more evidence supports good performance of predictive models. Kappa values between 0.3 and 0.5 denotes moderate agreement.

3.4.2.3 ROC Curve

For two-class cases, additional statistics can be added to evaluate the model. Sensitivity is the percentage of event predicted correctly for samples having the event, i.e. true positive rate. Similarly, specificity is defined as the percentage of nonevent predicted correctedly for samples not having the event.

$$Sensitivity = \frac{\text{Number of TP}}{\text{Number of (TP + FN)}} = \text{true positive rate}$$

$$Specificity = \frac{\text{Number of TN}}{\text{Number of (TN + FP)}} = \text{1 - false positive rate}$$

Obviously, there is a trade-off between sensitivity and specificity of the model. If more positive predictions are made, then sensitivity will increase and specificity will decrease. In bank marketing, contacting more clients that will not subscribe long-term deposit is acceptable for business. Since there is a trade-off, it is important to carefully choose an effective threshold for continuous prediction output within (0,1) to determine class prediction. Receiver operating characteristic (ROC) curve is generated to evaluate the trade-off and can be summarized into one single value.

For logistic regression model, the default threshold for probability is 50%. In problem of interest, decreasing the threshold will capture more clients willing to subscribe the deposit but at the meantime incorporating more clients that not. ROC curve is created by illustrating (1 - specificity, sensitivity) combination for each possible threshold. That is, plot true-positive rate against false-positive rate. A good ROC curve has steep trajectory for left part of the curve. A baseline for comparion is 45° diagonal line. Curve above that line means increasing rate of sensitivity is larger than decreasing rate of specificity, which is a good sign.

ROC curve is helpful for determining a propriate threshold. Moreover, it further proposes a quantitative measurement for model performance. A perfect ROC curve would be a horizontal line from (0,1) to (1,1), under which the area is 1. A random guessing with no model would be a 45° diagonal line, under which the area is 0.5. An effective predictive model will exhibit a ROC curve between the two types mentioned above with area within (0.5,1). Area under ROC curve is termed as AUC. Even for different predictors or models, AUC is an easy assessment of model as well as model comparison. The more the curve is shifted toward (0,1), the better the model is. To translate it into AUC: the larger the AUC, the better the model is. Confidence interval can also be calcuated for AUC. Sometimes we focus on partial ROC curve if event of interest cares more about a specific part.

3.4.2.4 Lift Charts

When ordered by event probability, we would expect to see samples with events are ranked higher than those without events. Here comes the idea of lift charts. They first rank samples with their event probability and then generate cumulative event rate as more samples are taken. In a perfect world, the m highest ranked samples would be m samples with events. By randomly selecting samples, lift is the number of samples with events detected.

Lift charts are constructed first by applying predictive model to test data and obtain estimated event probability. Then order the samples by the classification probability obtained in first step. For each unique classification rate, select samples with estimated probability under that threshold. Within samples selected, calculate the true proportion of events as baseline event rate. To obtain percentage of events detected, divide classification rate by baseline event rate.

Lift chart plots cumulative lift against cumulative percentage of samples, which indicates efficiency of targeting clients in marketing scenario. Like ROC curve, reference line is also the 45° diagonal line. In addition, it applies to different models and can be used to compare models. Like AUC, area under lift charts is an alternative way to measure model performance.

Chapter 4

Model Implementation and Comparison

4.1 Model Implementation

4.1.1 Random Forests

As shown in Algorithm 1, Number of predictors to select at each split m_{try} is an important parameter. This distinguishes random forests from other tree-based ensembles. Number of trees to build m is another parameter to tune. As random forest is unlikely to overfit as other classification trees, m_{try} will be the primary tuning target and m the secondary.

We adopt function *tuneRF* in R package *randomForest* to tune m_{try} with 1000 trees. The optimal m_{try} is chosen by smallest OOB error (Table 4.1). As OOB error reaches the lowest point at $m_{try} = 2$, number of predictors tried at each split is set as 2. The same result when number of trees is 500.

Table 4.1: Tuning m_{try} for random forests

m_{try}	1	2	4	8
OOB Error	0.1008	0.0990	0.1016	0.1071

With optimal m_{try}, number of trees to grow is tuned at (1000, 1500, 2000, 3000) and respective OOB errors are (9.96%, 9.97%, 9.95%, 9.95%). Number of trees is set as 3000. Implementing function *randomForest* to training data, we obtain random forests model and corresponding variable importance for bank marketing data. With *predict* function, labels for test data are predicted and combined with real labels for accuracy rate, ROC curve and lift chart. Model performance evaluation of three models will be discussed together in Section 4.2.

4.1.2 Gradient Boosting

As shown in Algorithm 2, three parameters - different values of iterations M, interaction depth and shrinkage rate λ - need tuning. Combinations of M = (500,1000,1500,2000,3000), interaction depth = (1,2) and $\lambda = (0.01, 0.05, 0.1)$ are explored using *train* function to discover optimal parameter combination. 200 boostrap samples are generated during tuning and out-of-bag fraction is 0.5. Accuracy and Kappa are obtained to choose optimal model in Table 4.2.

Interaction depth 1 means only 1 predictor variable will be used to define partition at each node. Depth at 1 or 2 often is adequate if number of trees is large enough. The more interactions, more complicated the trees are. Shrinkage rate (i.e. learning rate) defines the discount for each learning step. With increasing learning rate, learning step is bigger but at risk of missing the optimal value in between. On the other hand, if learning rate decreases, predictions will move slowly and number of trees needs to increase by inverse proportion.

When interaction depth is held at 1 and shrinkage rate is relatively

Table 4.2: Tuning parameters for stochastic gradient boosting

Interaction depth	Shrinkage rate λ	Number of trees	Accuracy	Kappa
1	0.01	500	0.8992	0.2666
1	0.01	1000	0.8994	0.2727
1	0.01	1500	0.8996	0.2770
1	0.01	2000	0.8997	0.2807
1	0.05	500	0.8996	0.2838
1	0.05	1000	0.8997	0.2934
1	0.05	1500	0.8998	0.2985
1	0.05	1500	0.9003	0.2998
1	0.05	2000	0.8998	0.3016
1	0.05	2000	0.9003	0.3028
1	0.05	3000	0.9002	0.3068
1	0.1	500	0.9000	0.2956
1	0.1	1000	0.9000	0.3028
1	0.1	1500	0.8998	0.3058
1	0.1	2000	0.8998	0.3109
2	0.01	1000	0.8999	0.2818
2	0.01	3000	0.9006	0.3149
2	0.05	1000	0.9005	0.3250
2	0.05	3000	0.8992	0.3349
2	0.1	500	0.9003	0.3246
2	0.1	1000	0.8997	0.3316

large (0.1), 500 trees often give better accuracy rate and Kappa. That less trees are required because learning step is very big . When shrinkage rate is relatively small (0.05 or 0.01), either with interaction depth 1 or 2, more trees is always preferrable in terms of high accuracy and large Kappa.

As we discussed in Subsection 3.4.2.2, accuracy as a standard will bias us to value more about misclassification other than node impurity for classification trees. To reduce bias, Kappa statistic incorporates no-information rate as baseline. Therefore, Kappa statistic is used to determine the optimal boosting model with largest value. To fit training data, interaction depth (tree depth) is set as 2, which allows 2 predictors at each node to define a rule for split. Shrinkage $\lambda = 0.05$ and number of iterations (number of trees) M = 3000.

Function *gbm* in R package *gbm* is applied to build stochastic gradient boosting with optimal parameters above to training data. It specifies 3 cross-validation folds and Bernoulli distribution because of 2-class response. The same way as random forests, labels for test data are predicted and combined with real labels for accuracy rate, ROC curve and lift chart. Model performance evaluation of three models will be discussed together in Section 4.2.

4.1.3 Logistic Regression

Besides attributes of client and campaign, 5 social and economic context attributes are included to predict subscription. These are employment variation rate, consumer price index, consumer confidence index, Euribor 3-month rate and number of employees. Among them, first attribute is likely to highly correlates to the last one. Two consumer indices may have a strong relationship. Correlation visulation is shown in Figure 4.1.

Three pairs - employment variation rate with consumer price index, Euribor 3-month rate and number of employees - have strong correlaions above 0.8. This suggests us to use PCA to resolve multicollinearity and reduce

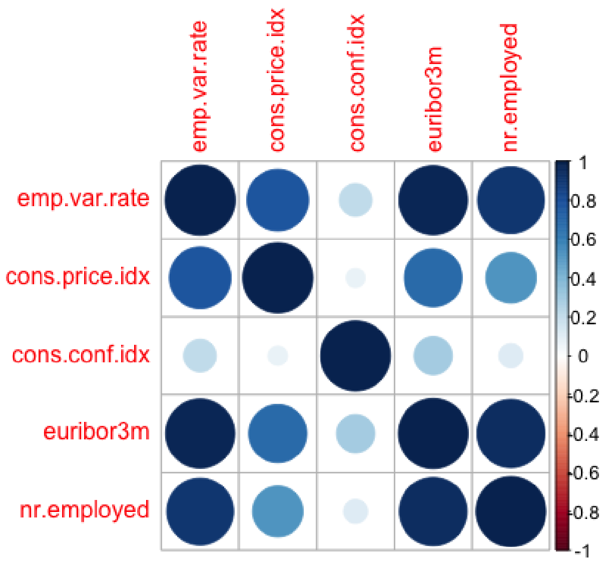

Figure 4.1: Correlation visualization for 5 social and economic variables

dimension. PCA output is summarized below. PC1 - PC5 denotes principal components generated in sequence.

Table 4.3: PCA summary for social and economic attributes

	PC1	PC2	PC3	PC4	PC5
Standard deviation	1.8659	0.9862	0.7139	0.15916	0.10411
Proportion of Variance	0.6963	0.1945	0.1019	0.00507	0.00217
Cumulative Proportion	0.6963	0.8909	0.9928	0.99783	1.00000

Cumulative variance proportion of top 3 components reaches 99.28%. They will replace the original 5 attributes in the model for logistic regression.

In addition to data pre-processing, binning some predictors is applied to generate more significance and interpretability. For example, we would not expect a 32-year-old man behave totally different than a 33-year-old man when it comes to subscription of bank deposit. We categorize *age* as 'Young'(below 25), 'Adult'(25-60) and 'Senior'(60 above); *compaign* as 'contact <= 3' and 'contact > 3'; *pdays* as 'new client' and 'previous client'; *previous* as 'no contact before', 'contact once before' and 'contact more than once before'; *month* into four seasons; *job* as 'office job', 'physical worker', 'flexiable worktime job', 'student' and 'retired'.

Fit *glm* to data with recategorized predictors, we found many predictors insignificant. We then apply backward selection to remove predictors having limited impact on outcome. Variables *marital, education, loan, previous* are removed. Hosmer and Lemeshow Goodness of Fit test shows Chi-square = 22.3969 with p-value 0.004231. This indicates the logistic regression after backward selection does not fit the data very well.

To improve model fitting, we rebin *day of week* into 'Monday' and 'non-Monday' as non-Mondays are significantly different than Monday. Similarly, recategorize *job* as 'not working' and 'working'; *default* into 'unknown' and 'known'; *month* as 'Spring & Fall' and 'Summer & Winter'. Again fit logistic regression without interactions using *glm* and perform a backward selection. Variables *housing, pdays* are no longer in the model. However, we still reject Goodness of Fit test with p-value 0.0072.

Then two-way interactions are added into the model to allow more nonlinearity. Almost all model terms are significant so no need to perform selection. It also passes Goodness of Fit test with p-value 0.25. This is the final logistic model. Predictions on test data are made and compared with those from classification trees in the following section.

From final logistic model, we extract first order results in Table 4.4. As coefficient of *contact(telephone)* is 1.30720 greater than 0 so contacting clients via telephone will have much more chance of success. The same way with *contact*, compaigning on non-Mondays is more likely to get positive response. If previous marketing campaign is successful or nonexistent, it is more likely to win this contact than when previous outcome is failure. This makes sense considering client loyalty and targeting. Senior compared to young people are easy to subscribe bank deposit may be due to more savings. Adult is less unlikly to buy than young people since financial burden can be largest for this group of people. It also suggests that campaigning during Summer and Winter is better.

4.2 Model Interpretation and Comparison

4.2.1 Variable Importance Interpretation

Variable importance from random forests and gradient boosting can be directly obtained after model fit. Two types of variable importance in random forests are given using *varImpPlot*. Measured by mean decrease accuracy and mean decrease Gini give similar results. *Age, job, euribor3m* are among top 5 important variables in both ways. Five social and economic attributes have strong or moderate impact on outcome. This indicates that general economic situation will affect personal decision on investment. *Marital, loan, default, housing* have relatively less impact on outcome. Since Gini index focuses more

Table 4.4: First-order coefficient estimates of logistic regression

Coefficients	Estimate	Standard Error	Z value
Intercept	-2.42943	0.26775	-9.073
default(unknown)	-0.32355	0.06707	-4.824
contact(telephone)	1.30720	0.30899	4.230
day of week(non-Monday)	0.37213	0.07091	5.248 3
poutcome(nonexistent)	0.64674	0.07702	8.397
poutcome(success)	1.32441	0.24240	5.464
PC1	0.48655	0.03080	15.795
PC2	0.42876	0.11372	3.770
PC3	0.10389	0.11689	0.889
age(Adult)	-0.12301	0.09221	-1.334
age(Senior)	0.27996	0.14450	1.937
campaign(contact > 3)	-1.27904	0.25508	-5.014
pdays(previous client)	-0.77422	0.22757	-3.402
month(Summer & Winter)	0.82584	0.18375	4.494
job(other jobs)	-0.35926	0.08593	-4.181

on node purity, the right plot in Figure 4.2 will be used for comparison in the following.

Gradient boosting also measures variable relative importance (Figure 4.3) but different than random forests in details. Though we cannot compare the importance scores from two models, two rankings also provide some consistency. Both *nr.employed* and *euribor3m* rank the top 2 important variables but in opposite order. Both top 1 important variables are far more important than others. As it is discovered in correlation matrix before logistic regression, 2 variables have correlation above 0.9. It is not surprising to see they switch ranks in different models like substitutes. *Job, pdays, age, poutcome* rank moderately strong in both models. However, *month* is not as important as it

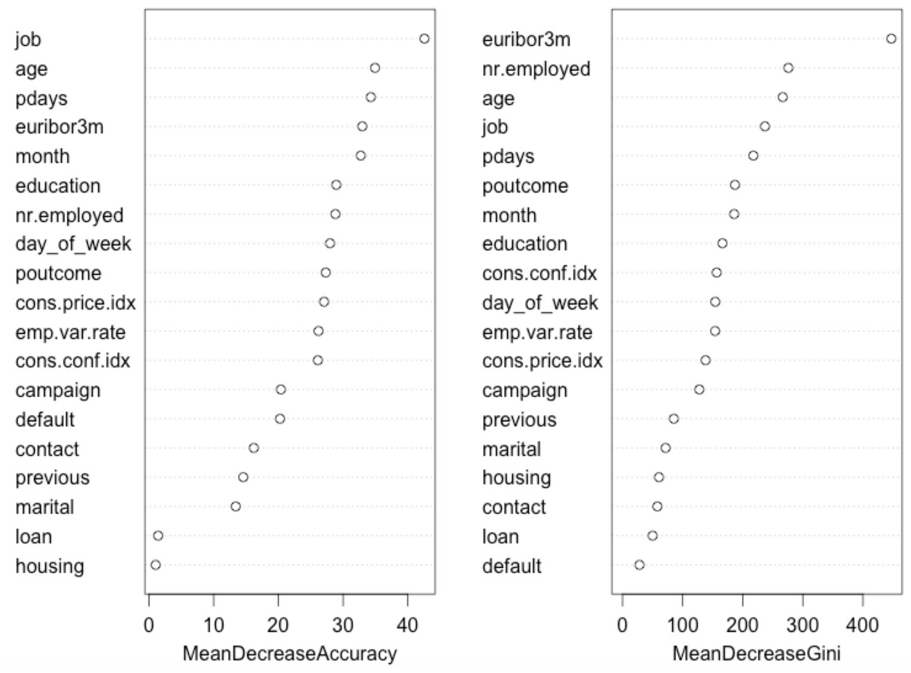

Figure 4.2: Variable importance for random forests with two measures

is in boosting. *Marital, housing, default* are least important in both models.

Logistic regression does not provide direct variable importance but Z-score can serve as an indicator for variable importance (Table 4.4). The larger the Z-score, the more significant the variable is. Coefficient estimates are not reliable since it is against baseline level and not comparable between variables. Z-value is the standardized version of coefficient estimate. Z value is given at level of categorical variable but not category as a whole so rough comparison to classification trees is summarized. Not surprisingly, PC1 with Z-value 15.795 is the most important variable. This is in alignment with two tree-based models where all original attributes rank strong or moderate impor-

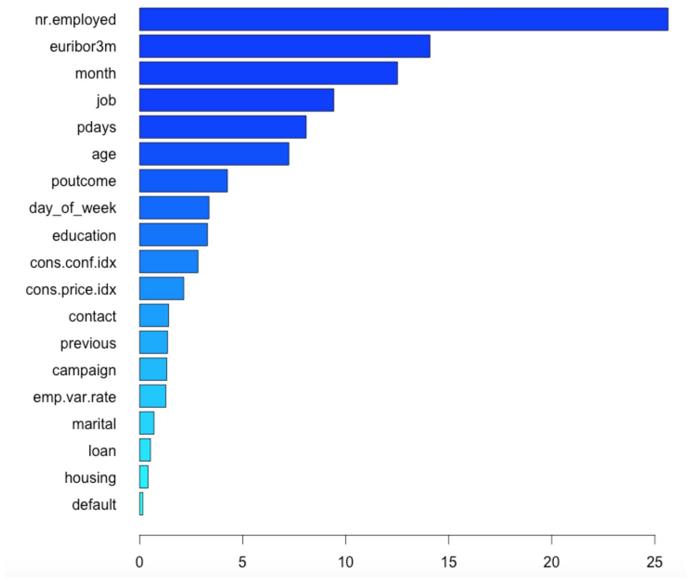

Figure 4.3: Variable relative importance for gradient boosting

tant. *Marital, housing* are removed by backward selection and also rank low in random forests and gradient boosting. *Month* is recategorized into season in logistic regression and become significant unlike the orignal variable in tree-based models. *Poutcome, day of week* are among top 3 significant variables in logistic regression. They are also moderately important variables in other 2 models. *Compaign* in 2 models is not as important as it is in regression.

Overall, variable importance remain consistent among 3 models though models deal with variables in different ways. Social and economic variables are among top important variables in 3 models. *Marital, housing, loan* are least important in all models.

4.2.2 Predictive Performance Comparison

Accuracy rate is plot against cutoff of classification for 3 models in Figure 4.4. As cutoff is increasing, random forests first reaches the plateau around cutoff 0.36 and then gradient boosting around 0.4. It takes longer for logistic regression to stablize. In addition, there is a jump in accuracy rate at cutoff 0.18 and 0.19.

ROC curve is a good visualization for model performance and area under ROC curve (AUC) is a good quantitative evaluation for model performance. In Figure 4.5, RF, GBM and LR represent random forests, gradient boosting model and logistic regression respectively. Values in the parentheses are corresponding AUC. By looking at ROC curves, it is a litte difficult to distinguish them except that RF curve falls below when false positive rate $>$ 0.2. GBM curve is above LR curve by a tiny amount. In the part where cutoff $<$ 0.2, 3 curves overlap with each other. However, gradient boosting has the largest value AUC 0.791, then LR 0.788 and RF 0.777. There is approximate 0.11 difference in between.

Lift charts delineate the effectiveness to capture positive reponse. It is correlated to business cost to reach targeted cutomers in marketing compaign. Pattern for 3 lift charts exhibit similar to ROC curves: GBM is slightly better than LR and RF, and RF falls below in the right part.

Overall, 3 models have similar predictive model performance in terms of ROC curves and lift charts. Best AUC 0.791 is given by gradient boost-

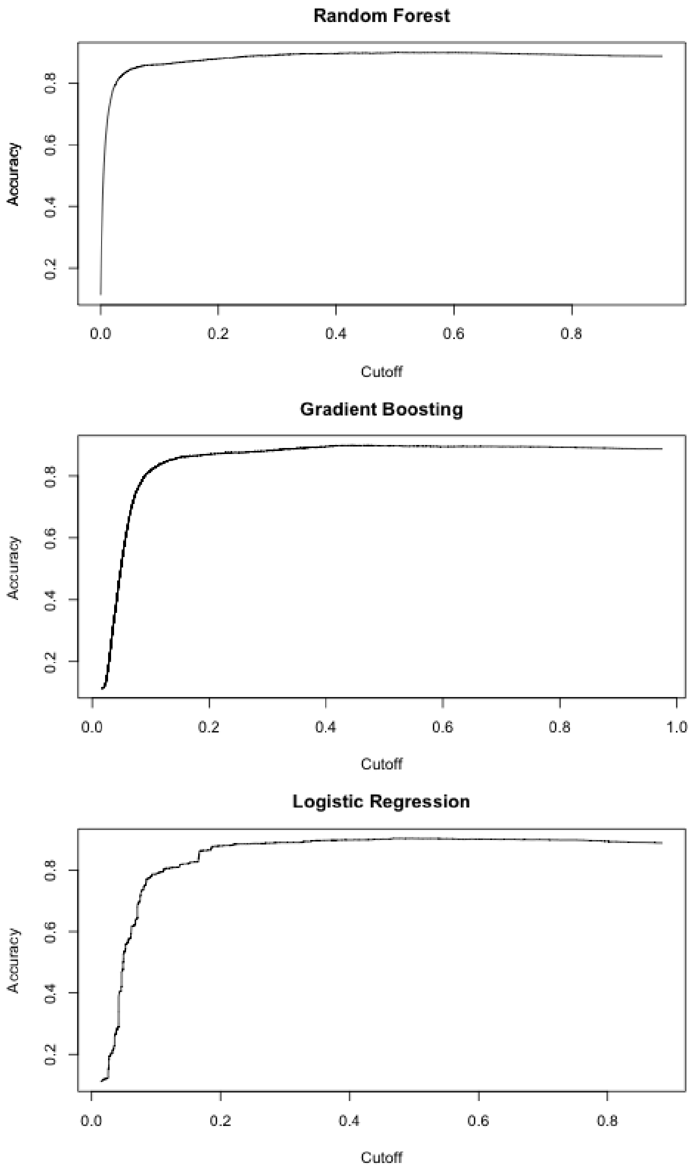

Figure 4.4: Test accuracy for three models

ing, which is close to AUC 0.794 provided by best model Neural Network[10]. Logistic regression after PCA and backward selection not only gives good pre-

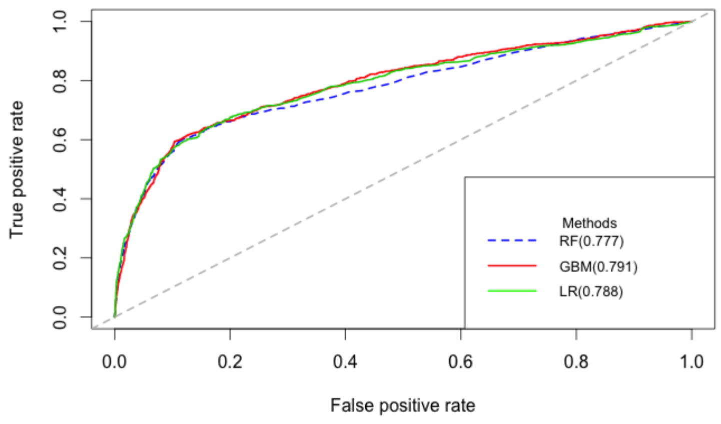

Figure 4.5: ROC curves for three models

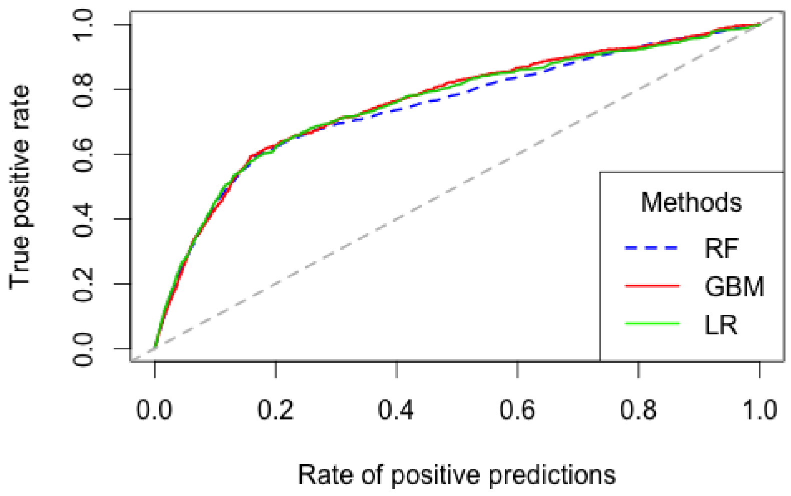

Figure 4.6: Lift charts for three models

dictions but also provide more actionable business insights such as better to compaign on non-Monday. Random forests and gradient boosting as tree-based ensembles get rid of overfitting by tuning number of trees and other parameters, and generate good results because of more flexibility.

Appendix
Variable Description

Variable	Description	Type
Age	age of client	Numerical
Job	type of client's job	Categorical
Marital	client's marital status	Categorical
Education	highest education level of client	Categorical
Default	Does client has credit?	Categorical
Housing	Does client has housing loan?	Categorical
Loan	Does client has personal loan?	Categorical
Contact	contact type of client	Categorical
Month	month of last contact with client	Categorical
Day of Week	day of the week for last contact	Categorical
Duration	duration of contact	Numerical
Campaign	number of contacts performed during this campaign	Numerical
Pdays	number of days from last contact in a previous campaign	Numerical
Previous	number of contacts performed before this campaign	Numerical
Poutcome	outcome of previous campaign	Categorical
Emp.var.rate	employment variation rate	Numerical
Cos.price.idx	consumer price index	Numerical
Cons.conf.idx	consumer confidence index	Numerical
Euribor3m	Euribor 3 month rate	Numerical
Nr.employed	number of employees	Numerical
y	Does the client subscribe a deposit?	Categorical

Bibliography

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

[3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[4] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[5] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[6] Ian Jolliffe. *Principal component analysis.* Wiley Online Library, 2002.

[7] Max Kuhn and Kjell Johnson. *Applied predictive modeling.* Springer, 2013.

[8] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[9] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems*, pages 431–439, 2013.

[10] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[11] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *MLDM*, pages 154–168. Springer, 2012.

[12] Greg Ridgeway. Generalized boosted models: A guide to the gbm package. *Update*, 1(1):2007, 2007.

[13] Chakarin Vajiramedhin and Anirut Suebsing. Feature selection with data balancing for prediction of bank telemarketing. *Applied Mathematical Sciences*, 8(114):5667–5672, 2014.