Classification to Predict Bank Telemarketing Success

Shuhui Guo

Contents

1 Introduction

Marketing campaigns become popular strategies for banks to improve their business performance. The campaigns are based on phone calls. Companies contact to clients directly to introduce their products like the bank term deposit. In the contemporary world, the phone calls could be made in several ways, such as telephone and cellular. Usually more than one calls need to be used to contact clients to access if the product would be subscribed. To avoid inefficient work unnecessary disturbance, figuring out how to identify the clients' willingness to subscribe the product is becoming important in telemarketing.

In this study, three machine learning algoriths on a marketing dataset are implemented to find the relationships between multiple attributes and the final outcome. Then the predictions could be made. Therefore, the companies could develop strategies and target those who have suitable needs to subscribe the products. Hopefully this study will give some suggestions to this field.

2 Data

2.1 Data Source

Data analysis in this report is based on the dataset **bank-additional-full.csv** on UCI Machine Learning Repository. This dataset is related with marketing campaigns of a Portuguese banking institution. The data contains all samples(41188) and 20 input variables from May 2008 to November 2010. The goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

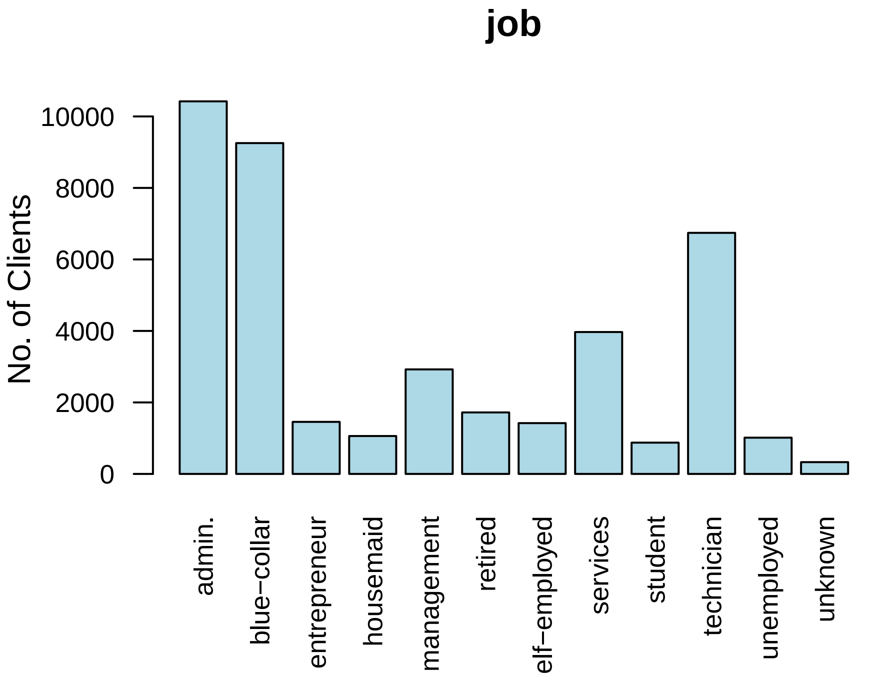
2.2 Data Preparation and Description

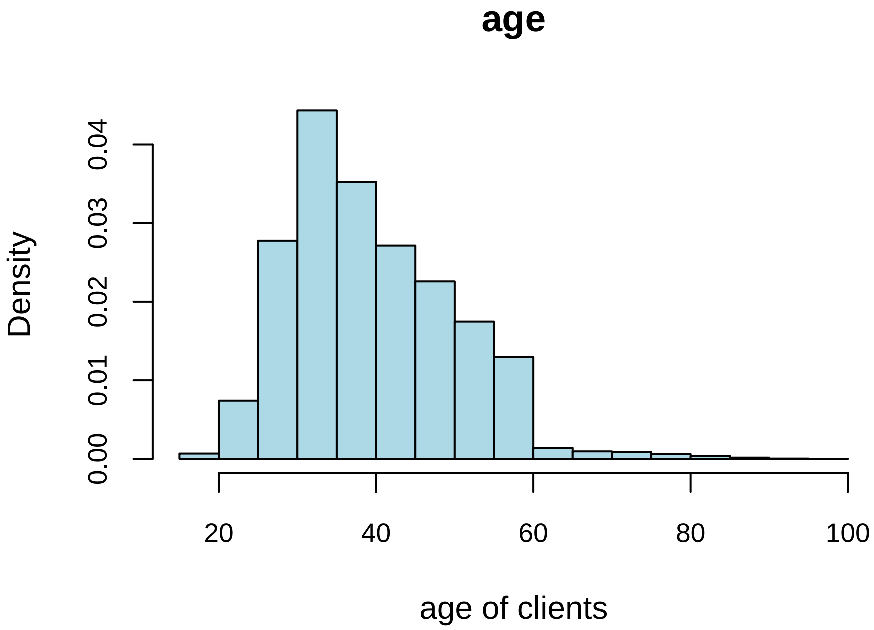
Since there appear to be no missing values, the total number of observations is 41188. For the input variables, there are 10 numerical variables and 10 categorical variables. For model fitting and realistic prediction, the input variable **duration** should be discarded because it highly affects the outcome. Also, the input variale **nr.employed** is rounded because the observations in this variable should be integers while there exists decimals in the data. Finally, the dataset used for analysis includes 41188 observations, 19 input variables and 1 output variable.

The input variables are shown as below:

	variable	data type	explanation
bank client data:			
1	age	numerical	Age at the contact date
2	job	categorical	job type
3	marital	categorical	marital status
4	education	categorical	education level
5	default	categorical	has credit in default?
6	housing	categorical	has housing loan?
7	loan	categorical	has personal loan?
the last contact of current campaign:			
8	contact	categorical	contact communication type
9	month	categorical	last contact month of year
10	day of week	categorical	last contact day of the week
other attributes:			
11	campaign	numerical	number of contacts performed in this campaign and for this client
12	pdays	numerical	number of days passed by after the client was last contacted in a previous campaign
13	previous	numerical	number of contacts performed before this campaign and for this client
14	poutcome	catagorical	outcome of the previous marketing campaign
social and economic attributes:			
15	emp.var.rate	numerical	employment variation rate - quarterly indicator
16	cons.price.idx	numerical	consumer price index - monthly indicator
17	cons.conf.idx	numerical	consumer confidence index - monthly indicator
18	euribor3m	numerical	euribor 3 month rate - daily indicator
19	nr.employed	numerical	number of employees - quarterly indicator

To get an understanding of the data, the distributions of outcomes by **job** and **age** are visualized as below.


job
No. of Clients
0
2000
4000
6000
8000
10000
admin.
blue−collar
entrepreneur
housemaid
management
retired
elf−employed
services
student
technician
unemployed
unknown


age
Density
0.00
0.01
0.02
0.03
0.04
20
40
60
80
100
age of clients

3 Modeling

To build models, I first divide data into training and testing datasets. 75% of the observations are splitted into the training dataset and the remaining 25% are splitted into the testing dataset.

Then I will try Logistic Regression, Classification Tree and Gradient Boosted Machine to create models and do prediction.

3.1 Logistic Regression

In this case, there are two possible outcomes, *Yes* and *No*. Denoting the probability of these two outcomes as $P(Y)$ and $P(N)$ respectively, the probability of *Yes* could be written as $P(Y) = p$, and the probability of *No* could be written as $P(N) = 1 - p$. Thus, the odds of the *Yes* outcome, which is the ratio of the probability of *Yes* to the probability of *No*, is given by the following expression:

$$Odds(Y) = \frac{p}{1-p}$$

The logit function can be written as:

$$logit(p) = log(\frac{p}{1-p}) \quad where \quad 0 \leq p \leq 1$$

The logistic function is the inverse of the logit function. The logistic function is defined for a parameter α by the following expression:

$$logistic(\alpha) = \frac{1}{1 + exp(-\alpha)}$$

In the regression model, the logistic function is applied to the dependent variable to make binary classification prediction. Thus, if we have the following model:

$$y = mx + b$$

the logistic regression model could be expressed as:

$$logistic(y) = \frac{1}{1 + exp(-y)}$$

3.1.1 Logistic Regression Model Fitting

Fit the above logistic regression model to the training data, and the coefficients are obtained. In the results, all other coefficients performs normally but the variable **loan-unknown**. NA is introduced in the results of **loan-unknown** so that this variale becomes meaningless in model fitting. To address this problem, the colinearity is checked by testing the significance of associations between **loan** and other variables. The conclusion is that the variable **loan** and the variable **housing** is highly correlated. Also, after checking the data, there appears that the observations whose $loan = "unknown"$ are the same as the ones with $housing = "unkown"$. Logistic regression model cannot solve this problem. Instead, it will just discard the variable **loan-unknown** while doing prediction. Therefore, there should be some improvements in the model fitting.

3.1.2 Logistic Regression with Lasso Penalty

The lasso penalty could be written as:

$$\hat{\beta} = \arg\ \min_{\beta} \frac{1}{2} ||y - X\beta||^2 + \lambda ||\beta||_1$$

where λ is the penalty level which could be selected by cross-validation.

With this method, the varaibles used for prediction are selected more precisely thus the problem of colinearity could be solved.

The selected variables and corresponding coeffitients are as below:

variable	coefficient	variable	coefficient
Intercept	44.932	job-retired	0.219
job-student	0.238	education-university.degree	0.067
default-unknown	-0.094	contact-telephone	-0.198
month-jul	0.033	month-mar	0.736
month-may	-0.662	month-nov	-0.122
day_of_week-mon	-0.170	day_of_week-wed	0.011
campaign	-0.008	pdays	-0.001
poutcome-nonexistent	0.193	poutcome-success	0.692
emp.var.rate	-0.071	cons.conf.idx	0.005
nr.employed	-0.009		

Take the variables **day of week-mon** and **poutcom-success** as example. For every one unit change in **day of week-mon** (Monday is the last contact day), the log odds of subscribing to a long-term deposit decreases by 0.170. This might be because the clients contacted in Monday have just returned to work or study, they have less consideration on subscribing to a deposit. For every one unit change in **poutcome-success** (successful outcome of previous marketing campaign), the log odds of subscribing increases by 0.692. This might be because a successful outcome in previous campaign is more likely to lead to success in the current campaign.

Set the threshold as 0.5. The prediction probabilities higher than 0.5 should be classified to *Yes*, and the probabilities lower than 0.5 should be classified to *No*.

The confusion matrix of the training dataset is constructed as below:

```
##      true
## pred     no   yes
##   no  27103  2770
##   yes   308   710
```

Based on this result, out of 30891 observations in total, the model classifies 27813 correctly, which is 0.900. This result seems to be impressed. Nevertheless, the classification accuracy of each outcome is quite different. Out of the 27411 observations which did not subscribe to a deposit, the model classifies 27103 correctly, which is 0.989. On the other hand, out of the 3480 observations which subscribed to a deposit, the model classifies 710 correctly, which is only 0.204.

Use the trained model to make prediction in testing dataset. The confusion matrix of the testing dataset is:

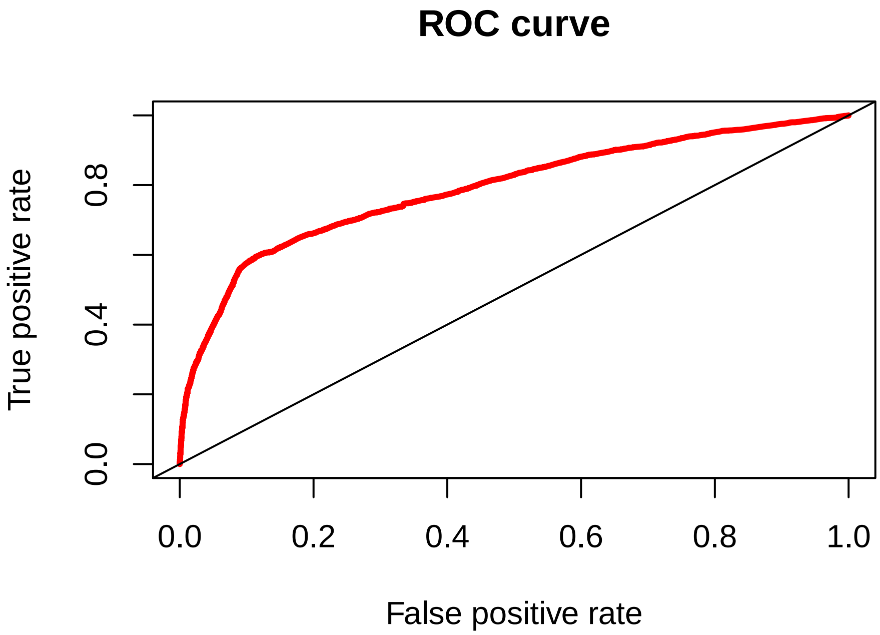
```
##      true
## pred    no  yes
##   no  9020  935
##   yes  117  225
```

Based on this result, the total classification accuracy is 0.898. But the unbalanced issue also exists. The classification accuracy of the observations which did not actually subscribe to a deposit is 0.987. While the classification accuracy of the observations which actually subscribed to a deposit is 0.194.

3.1.3 Model Improvement

In this part, the unbalanced issue will be addressed by checking ROC curve and finding the best threshold to do classification.

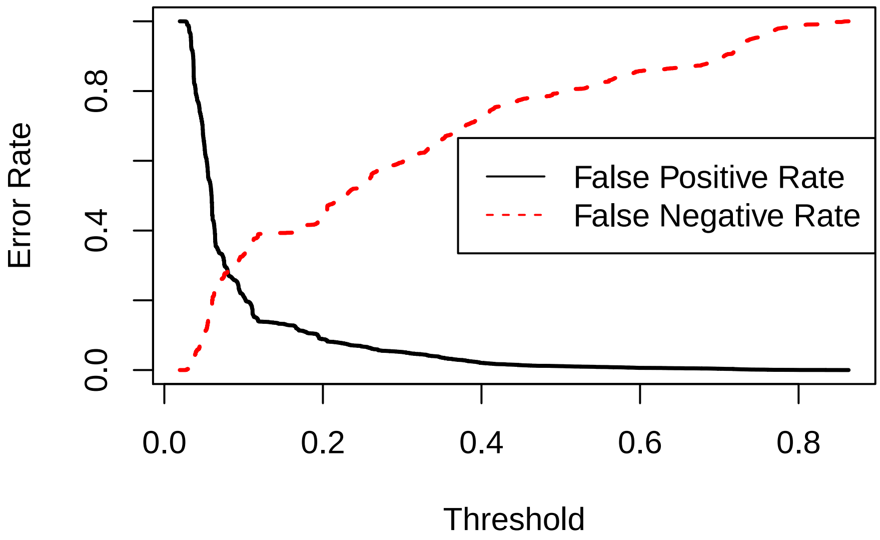
The ROC curve of training dataset is plotted as below:



The area under the curve (AUC) is used to summarize the performance of the model, and it is currently about 0.789. This value is pretty high, which indicates that the model has a good performance. Nevertheless, based on the ROC curve, the performance of the model could be better. The highest true positive rate in ROC curve is more than 0.900. While the true positive rate in current result is 0.204, which is much lower.

To make the classification results better, the false positive rate(fpr) and false negative rate(fnr) should be as small as possible. This could be achieved by changing the threshold, which is currently 0.5.

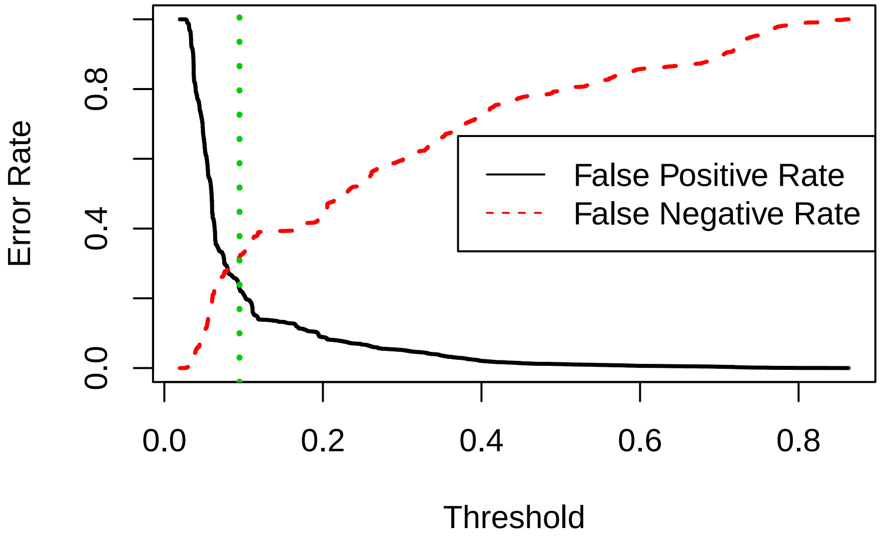
The changes of false positive rate and false negative rate with the changing of threshold is plotted as below:



To ensure the false positive rate(fpr) and false negative rate(fnr) are as small as possible, there should be a threshold in which the distance between (fpr,fnr) and (0,0) is the smallest. The Euclidean distance is used to measure this distance.

$$d = \sqrt{fpr^2 + fnr^2}$$

Finally the smallest distance is selected to be 0.391. The corresponding threshold is 0.094. The results are shown in the plot:



Based on the above inference, the prediction probabilities higher than 0.094 should be classified to *Yes*, and the probabilities lower than 0.094 should be classified to *No*.

The new confusion matrix of the training dataset is constructed as below:

```
##      true
## pred     no   yes
##   no  21234  1114
##   yes  6177  2366
```

Out of 30891 observations in total, the model classifies 23600 correctly, which is 0.764. This result is lower than before, but the classification accuracy of each outcome should also be checked. Out of the 27411 observations which did not subscribe to a deposit, the model classifies 21234 correctly, which is 0.775. On the other hand, out of the 3480 observations which subscribed to a deposit, the model classifies 2366 correctly, which is 0.680. The true positive rate is increased a lot and the overall model is much more accurate than before.

Use the trained model to make prediction in testing dataset. The confusion matrix of the testing dataset is:

```
##      true
## pred    no  yes
##   no  6992  354
##   yes 2145  806
```

The total classification accuracy of testing data is 0.757. The classification accuracy of the observations which did not actually subscribe to a deposit is 0.765. While the classification accuracy of the observations which actually subscribed to a deposit is 0.695.

3.2 Classification Tree

Classification tree is a flexible model to handle both categorical and continuous variables in a natural way. Also, it is robust to outliers. So in this part, the classification tree will be used to predict if the client will subscribe a deposit. Nevertheless, there is a tree size issue existing in the model fitting. A large tree (with many splits) can easily overfit the data. A small tree may not capture important structures. So to fit the model, the best size of tree should be decided.

3.2.1 Tree Size Selection

First fit the entire tree T_{max}. Then prune it to a sub-tree of T_{max} , denote as $T \preceq T_{max}$, which minimizes

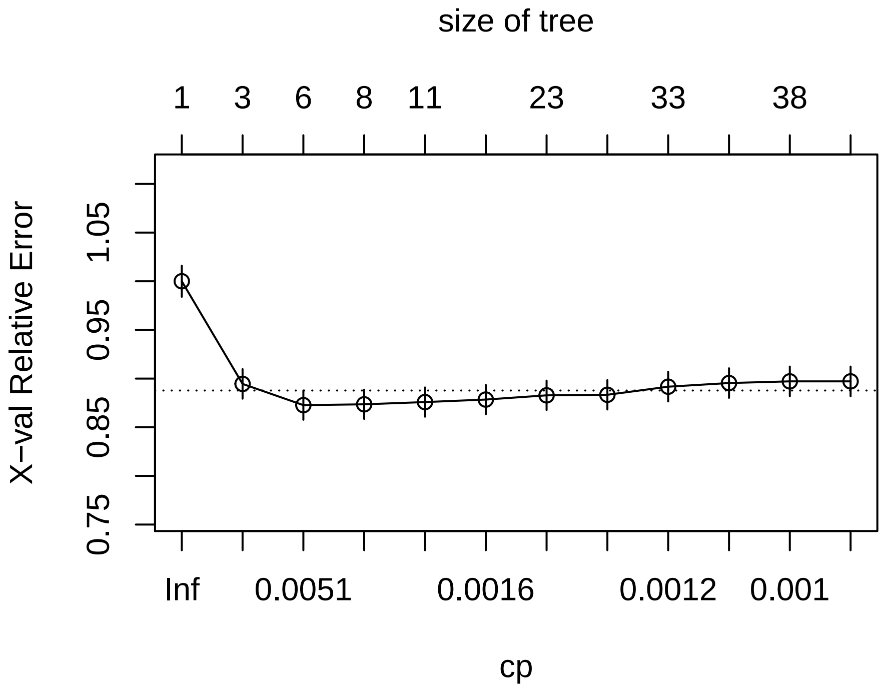
$$\begin{aligned} C_\alpha(T) &= \sum_{allterminalnodestinT} N_t \cdot Impurity(t) + \alpha \mid T \mid \\ &= C(T) + \alpha \mid T \mid \end{aligned}$$

where N_t is the number of observations in t, $\mid T \mid$ is the the number of terminal nodes of T, α is the complexity parameter chosen by cross-validation in the following steps:

Step1. Split the data randomly into 10 folds. Use 10-fold cross-validation and fits each sub-tree $T_1, ..., T_m$ on each training fold.

Step2. Calculate the corresponding misclassification risk R_m for each sub-tree and select the complexity parameter α giving the lowest misclassification risk.

The changes of misclasification risk with changes of α is:



Based on the above result, the complexity parameter α is selected to be 0.004 and the corresponding number of nodesize is 6.

3.2.2 Model Fitting and Prediction

The tree is shown in Figure 1.

Based on the above plot, we could see:

1. At the top node of the tree, for the variable **nr.employed**, 88% of the observations are higher than 5088, while 12% are lower than 5088.
2. Out of the observations with **nr.employed** higher than 5088, 93% did not subscribe the deposit and 7% subscribed the deposit.
3. For the observations with **nr.employed** lower than 5088, the next stop is to see how many of them with **pdays** higher than 16. 9% of the observations was last contacted in more than 16 days ago. 3% are less than 16 days and in this group, 28% did not subscribe the deposit while 72% did.
4. For the observations with **pdays** higher than 16, the next stop is to see whether the contact was by telephone. 1% of the observations was contacted by telephone and in this group, 78% did not subscribe the deposit while 22% did.
5. For the observations not contacted by telephone, the next stop is to see whether the variable **emp.var.rate** is lower than -2.4. There are 5% of the observations with

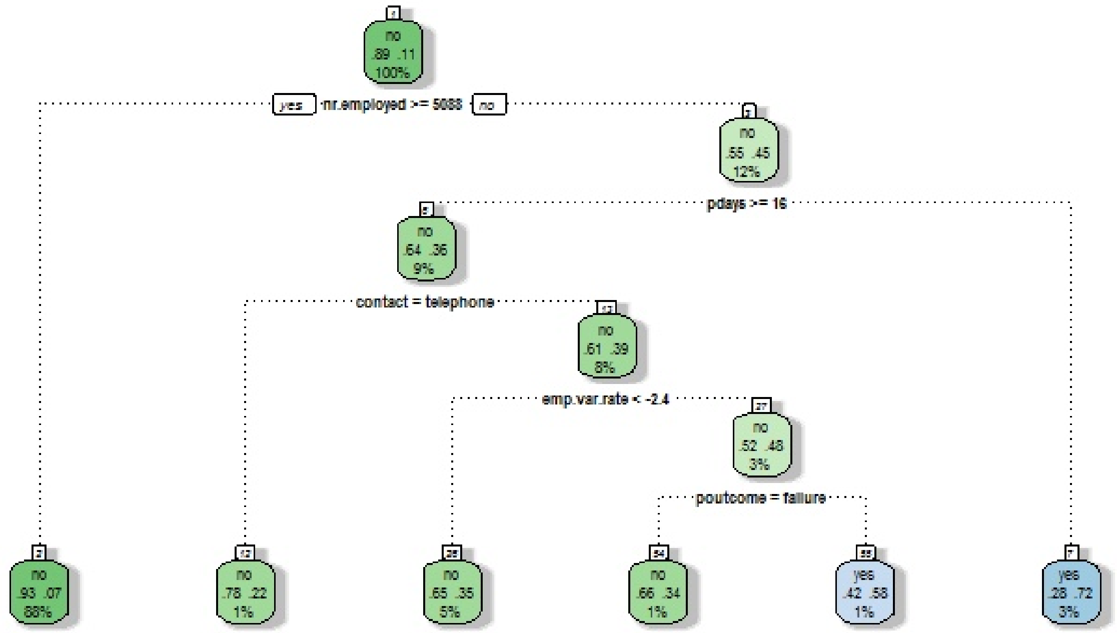


Figure 1: Classification Tree

emp.var.rate lower than -2.4. Out of these observations, 65% did not subscribe the deposit while 35% did.

6. For the observations with **emp.var.rate** higher than -2.4, the next stop is to see whether the outcome of the previous marketing campaign is failure. Out of the 1% of the observations whose previous outcome is failure, 66% did not subscribe the deposit while 34% did. Out of the 1% of the observations whose previous outcome is not failure, 42% did not subscribe the deposit while 58% did.

Use the trained model to make prediction in testing dataset. The confusion matrix is:

```
##       true
## pred    no  yes
##   no  8959  874
##   yes  178  286
```

The total classification accuracy of testing data is 0.898.

3.3 Gradient Boosted Machine

Gradient boosted machine is a machine learning technique used for regression and classification. In this part, gradient boosted machine will be used to predict if the client will subscribe a deposit.

3.3.1 Model Fitting

Fit the gradient boosted model with 1000 trees and the shrinkage parameter 0.01. Set the threshold of outcome as 0.5. The prediction probabilities higher than 0.5 should be classified to *Yes*, and the probabilities lower than 0.5 should be classified to *No*.

The confusion matrix of the training dataset is constructed as below:

```
##      true
## pred     no   yes
##   no  27092  2755
##   yes   319   725
```

Based on this result, out of 30891 observations in total, the model classifies 27817 correctly, which is 0.9. The total accuracy is pretty high. Nevertheless, the classification accuracy of each outcome is quite different. Out of the 27411 observations which did not subscribe to a deposit, the model classifies 27092 correctly, which is 0.988. On the other hand, out of the 3480 observations which subscribed to a deposit, the model classifies 725 correctly, which is only 0.208.

Use the trained model to make prediction in testing dataset. The confusion matrix of the testing dataset is:

```
##      true
## pred    no  yes
##   no  9008  922
##   yes  129  238
```

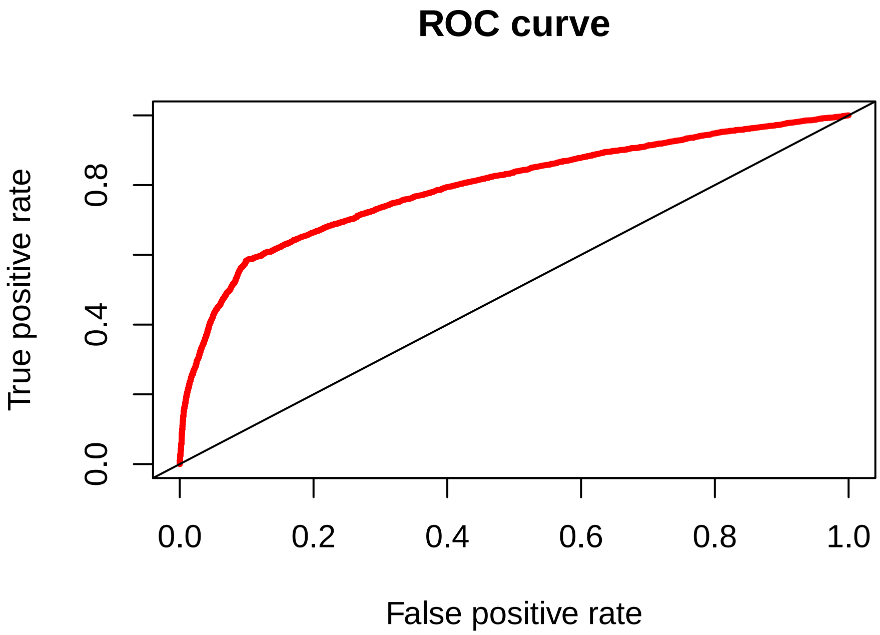
Based on this result, the total classification accuracy is 0.898. But the unbalanced issue also exists. The classification accuracy of the observations which did not actually subscribe to a deposit is 0.986. While the classification accuracy of the observations which actually subscribed to a deposit is 0.205.

Therefore, the ROC curve should be used to balance the prediction accuracy.

3.3.2 Model Improvement

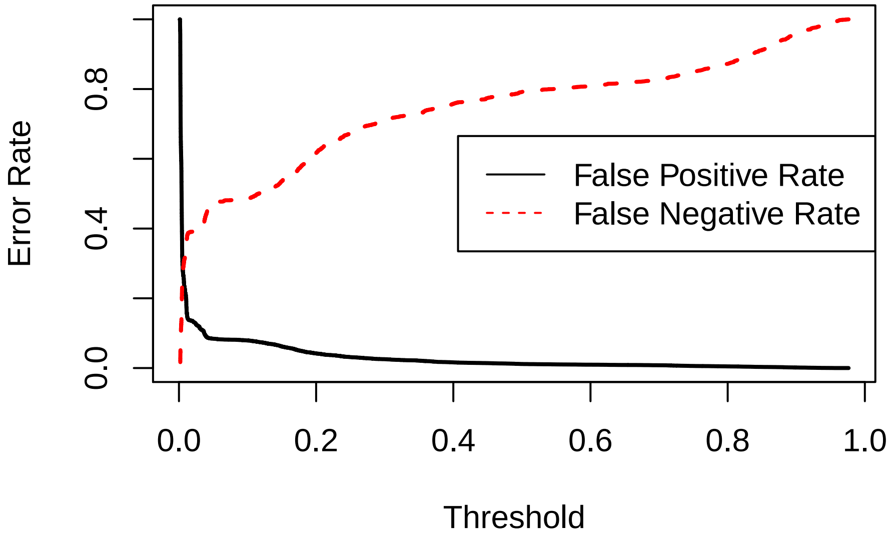
In this part, the unbalanced issue will be addressed by checking ROC curve and finding the best threshold to do classification.

The ROC curve of training dataset is plotted as below:



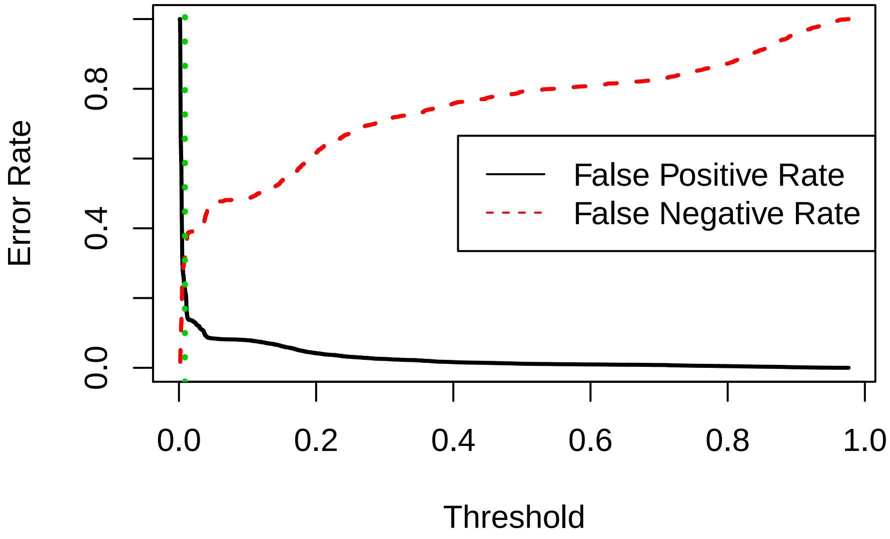
In this model, AUC is 0.793, which indicates that the model has a good performance. Nevertheless, based on the ROC curve, the performance of the model could be better.

To make the classification results better, the false positive rate(fpr) and false negative rate(fnr) should be as small as possible. This could be achieved by changing the threshold, which is currently 0.5. The changes of false positive rate and false negative rate with the changing of threshold is plotted as below:



To ensure the false positive rate(fpr) and false negative rate(fnr) are as small as possible, there should be a threshold in which the distance between (fpr,fnr) and (0,0) is the smallest, which is measured by the Euclidean distance.

Finally the smallest distance is selected to be 0.388. The corresponding threshold is 0.009. The results are shown in the plot:



Based on the above inference, the prediction probabilities higher than 0.009 should be classified to *Yes*, and the probabilities lower than 0.009 should be classified to *No*.

The new confusion matrix of the training dataset is constructed as below:

```
##       true
## pred     no   yes
##   no  21307  1105
##   yes  6104  2375
```

Out of 30891 observations in total, the model classifies 23682 correctly, which is 0.767. This result is lower than before, but the classification accuracy of each outcome should also be checked. Out of the 27411 observations which did not subscribe to a deposit, the model classifies 21307 correctly, which is 0.777. On the other hand, out of the 3480 observations which subscribed to a deposit, the model classifies 2375 correctly, which is 0.682. The true positive rate is increased a lot and the overall model is much more accurate than before.

Use the trained model to make prediction in testing dataset. The confusion matrix of the testing dataset is:

```
##       true
## pred    no  yes
##   no  7033  352
##   yes 2104  808
```

The total classification accuracy of testing data is 0.761. The classification accuracy of the observations which did not actually subscribe to a deposit is 0.77. While the classification accuracy of the observations which actually subscribed to a deposit is 0.697.

4 Model Comparison and Conclusion

So far, three models have been fitted to make predictions. The performance measurements for the three models is shown as below:

measurements	Logistic Regression	Classification Tree	Gradient Boosted Machine
test accuracy	0.757	0.898	0.761
AUC	0.789	0.622	0.793

Based on these results, classification tree ranks first in test accuracy. Nevertheless, this model causes unbalanced issue and its AUC ranks the last. Thus clsassification tree is not the best model. Gradient boosted machine has higher accuracy than Logistic regression and ranks first in AUC. Therefore, gradient boosted machine is the best model.

References

S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011. EUROSIS.

Reddy, A. (2016, June 19). Building Machine Learning Models. Retrieved from https://rpubs.com/arjunreddyt/190610

Widjaja, J. (2017, May 24). Classification to Predict Bank Marketing Success. Retrieved from https://github.com/jesswidjaja/DataMiningProject

Sandhu, S. (2017, April 25). Rpart, cross validation. Retrieved from https://stats.stackexchange.com/questions/275652/rpart-cross-validation

Zhang, C. (2016, December 23). Machine Learning on Bank Marketing Data. Retrieved from https://nycdatascience.com/blog/student-works/machine-learning/machine-learning-retail-bank-marketing-data/