IE 7275-DATA MINING IN ENGINEERING CASE STUDY REPORT

Group – 11

Student Names: Reema Yadav and Sakshi Patil

Executive Summary

The major source of revenue generation for Banks is by attracting the public into the bank. Considering this there is a huge competition among various banks. Advancement in media and technology has given banks the ability to advertise their product in an effective and efficient manner with maximum consumers buying their product. Due to the advancement in the data mining and machine learning techniques, banks are now adopting the data-driven strategies to market their products. The goal of this case study is to use supervised machine learning algorithms and build a classification model for the dataset and suggest better method for the marketing campaign. The goal is to predict whether a client will subscribe to a term deposit or no with the help of a few given dependent variables.

The dataset used in this case study is the real-world data of a Portuguese Bank's Marketing Campaign. The dataset contains 45,211 rows and 17 columns out of which 10 are categorical type and 7 are numeric type. There are 16 independent variables and one dependent variable "y". The categorical variables are converted in factors and the numeric variables were normalized. The data is preprocessed and the classification algorithms like Logistic regression, SVM, K-NN, Decision tree, Random Forest, Naïve Bayes and Neural Nets were used to build the models. Model accuracy and performance evaluation like ROC curve and confusion matrix are used to check the performance of each model.

Majority of the attributes were used in the constructing binary classification models. Among all the approaches, Logistic regression with accuracy of 89.94% was used as the best model. The selected model is powerful and easy to implement. Duration of call between the sales representative and the client is an important factor which decides whether the client will buy the product or no, this is known through the decision tree and random forest models. We recommend the bank's marketing team to focus of on the duration of the calls where the customers are spending more than 375 seconds as they are more likely to subscribe to term deposit. We also suggest that the representatives have a questionnaire which will keep the customer engaged, increasing the call duration. Thus increasing the marketing campaign success rate, in-turn increasing the banks revenue.

I. Background Information

Problem Statement:

The data used is related to direct marketing campaigns of Portuguese bank. Marketing campaigns are vital for all the banks to progress. The given data for marketing campaign is based on the phone calls. Based on the input of the client, goal is to predict if the consumer will subscribe (yes/no) a term deposit (variable y)

Goal of the case study:

The aim of the study is to analyze and highlight the importance of marketing in the banking sector and the significance of the phone calls in the bank. Our task is to implement the Machine Learning algorithms on the historical-data collected from the Portuguese Bank.

Possible Solutions:
Different Supervised Machine learning techniques can be applied over here like decision tree and Random forest. We will be using different kind of regressions on the dataset to build a prediction model to see whether a potential customer will buy the term deposit or no.

II. Data Exploration and Data Visualization

Data Exploration:

The bank dataset has 17 columns consisting of both numeric and categorical and 45,211 rows.

```
> # Importing the dataset
> bank_data <- read.csv("~/Downloads/bank (1)/bank-full.csv", sep=";", stringsAsFactors = FALSE)
> exp_bankdata <- read.csv("~/Downloads/bank (1)/bank-full.csv", sep = ";", stringsAsFactors = TRUE)
> head(bank_data,5)
  age          job marital education default balance housing loan contact day
1  58   management married  tertiary      no    2143     yes   no unknown   5
2  44   technician  single secondary      no      29     yes   no unknown   5
3  33 entrepreneur married secondary      no       2     yes  yes unknown   5
4  47  blue-collar married   unknown      no    1506     yes   no unknown   5
5  33      unknown  single   unknown      no       1      no   no unknown   5
  month duration campaign pdays previous poutcome  y
1   may      261        1    -1        0  unknown no
2   may      151        1    -1        0  unknown no
3   may       76        1    -1        0  unknown no
4   may       92        1    -1        0  unknown no
5   may      198        1    -1        0  unknown no
```

```
> #Checking the dimensions of the dataset
> dim(bank_data)
[1] 45211    17
```

We checked the structure of the data and observe that 7 columns are numeric and the other 10 columns are categorical.

```
'data.frame':   45211 obs. of  17 variables:
 $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
 $ job      : chr  "management" "technician" "entrepreneur" "blue-collar" ...
 $ marital  : chr  "married" "single" "married" "married" ...
 $ education: chr  "tertiary" "secondary" "secondary" "unknown" ...
 $ default  : chr  "no" "no" "no" "no" ...
 $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
 $ housing  : chr  "yes" "yes" "yes" "yes" ...
 $ loan     : chr  "no" "no" "yes" "no" ...
 $ contact  : chr  "unknown" "unknown" "unknown" "unknown" ...
 $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month    : chr  "may" "may" "may" "may" ...
 $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
 $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome : chr  "unknown" "unknown" "unknown" "unknown" ...
 $ y        : chr  "no" "no" "no" "no" ...
```

Summary of the data was found to get more insights about the data

```
      age             job              marital            education           default        
 Min.   :18.00   Length:45211       Length:45211       Length:45211       Length:45211      
 1st Qu.:33.00   Class :character   Class :character   Class :character   Class :character  
 Median :39.00   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
 Mean   :40.94                                                                              
 3rd Qu.:48.00                                                                              
 Max.   :95.00                                                                              
    balance         housing              loan             contact               day       
 Min.   : -8019   Length:45211       Length:45211       Length:45211       Min.   : 1.00  
 1st Qu.:    72   Class :character   Class :character   Class :character   1st Qu.: 8.00  
 Median :   448   Mode  :character   Mode  :character   Mode  :character   Median :16.00  
 Mean   :  1362                                                            Mean   :15.81  
 3rd Qu.:  1428                                                            3rd Qu.:21.00  
 Max.   :102127                                                            Max.   :31.00  
    month              duration         campaign          pdays          previous          poutcome        
 Length:45211       Min.   :   0.0   Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000   Length:45211      
 Class :character   1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000   Class :character  
 Mode  :character   Median : 180.0   Median : 2.000   Median : -1.0   Median :  0.0000   Mode  :character  
                    Mean   : 258.2   Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803                     
                    3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000                     
                    Max.   :4918.0   Max.   :63.000   Max.   :871.0   Max.   :275.0000                     
      y            
 Length:45211      
 Class :character  
 Mode  :character  
```

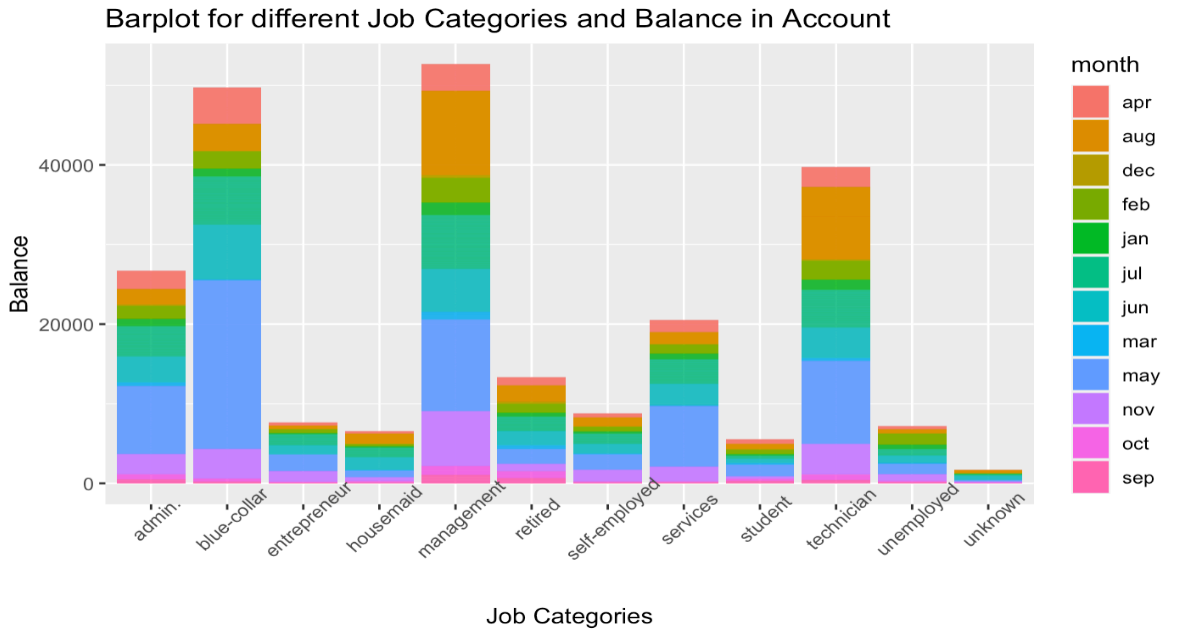
We can see that the numeric data columns are of different scales and hence have to normalized. This is done in-order to maintain the balance and no one variable will have more weightage in the classification.

Listing out the categorical variables in the dataset

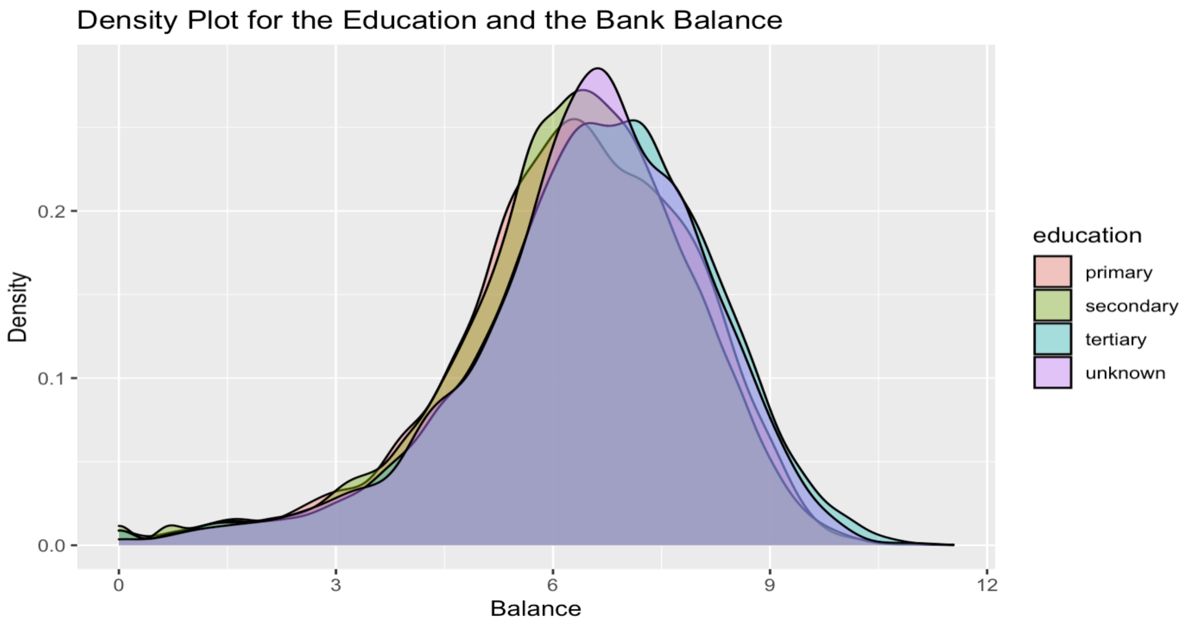
```
Listing out the various Categories
```{r}
levels(exp_bankdata$job)
levels(exp_bankdata$marital)
levels(exp_bankdata$education)
levels(exp_bankdata$default)
levels(exp_bankdata$housing)
levels(exp_bankdata$loan)
levels(exp_bankdata$contact)
levels(exp_bankdata$poutcome)
levels(exp_bankdata$y)
```

 [1] "admin."   "blue-collar"   "entrepreneur" "housemaid"   "management"   "retired"
 [7] "self-employed" "services"   "student"   "technician"   "unemployed"   "unknown"
[1] "divorced" "married" "single"
[1] "primary"   "secondary" "tertiary" "unknown"
[1] "no" "yes"
[1] "no" "yes"
[1] "no" "yes"
[1] "cellular" "telephone" "unknown"
[1] "failure" "other"   "success" "unknown"
[1] "no" "yes"
```

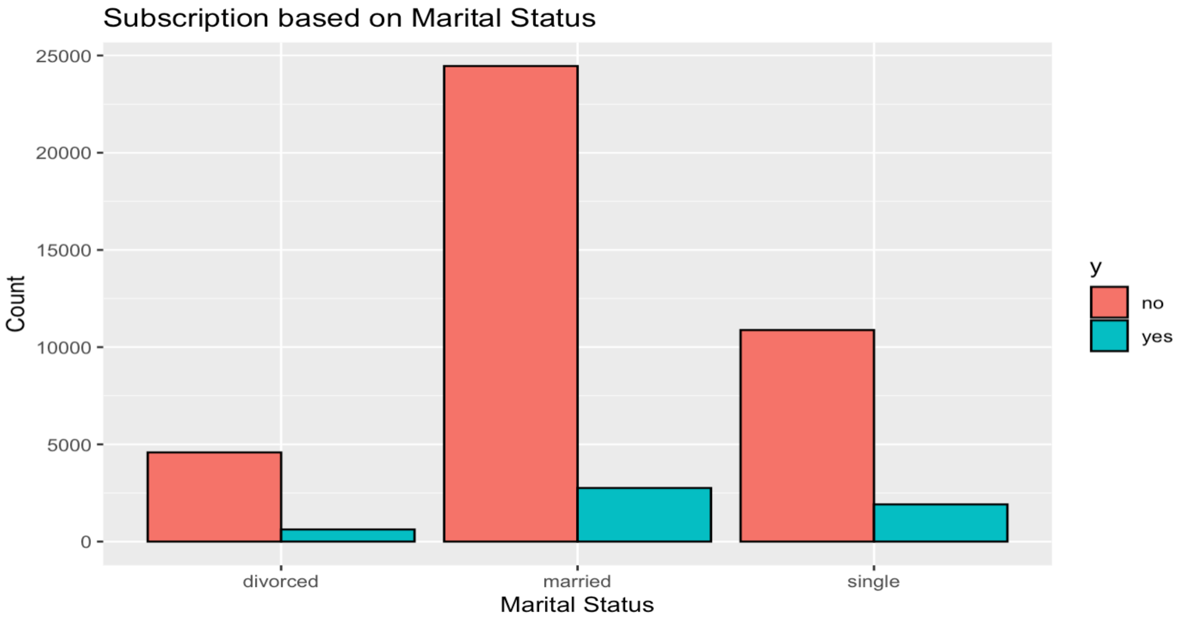
Data Visualization:



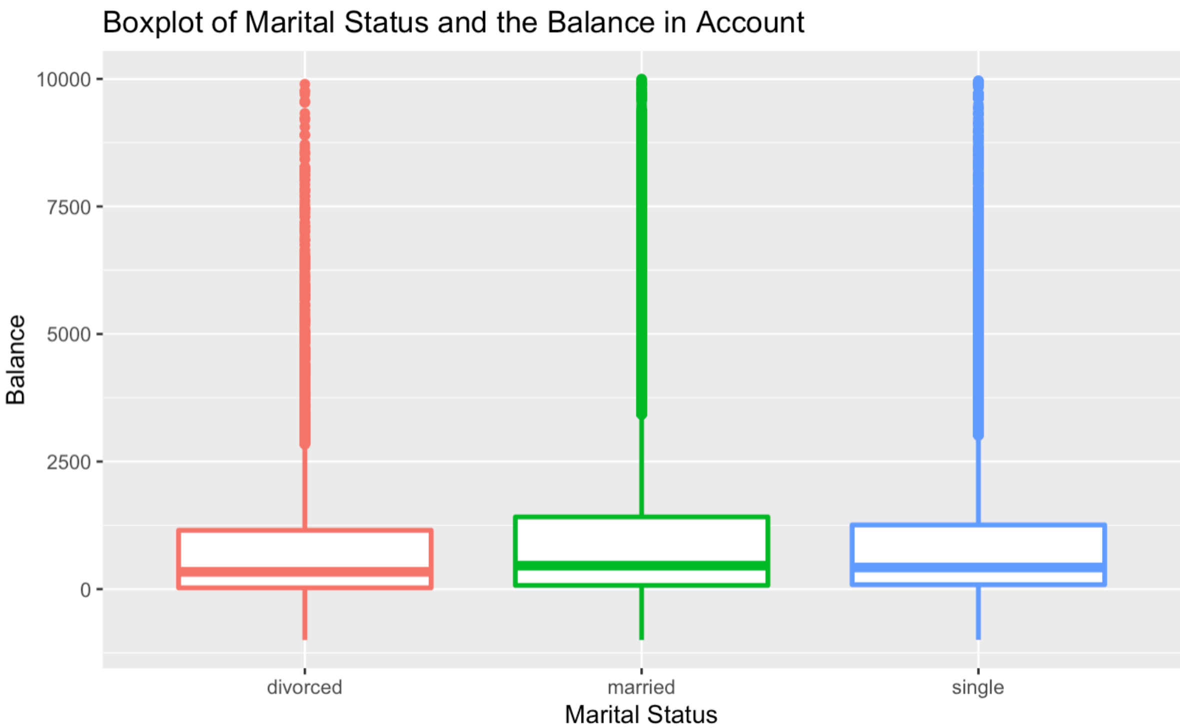
The above plot shows the bank balance for different job categories at different months of the year.



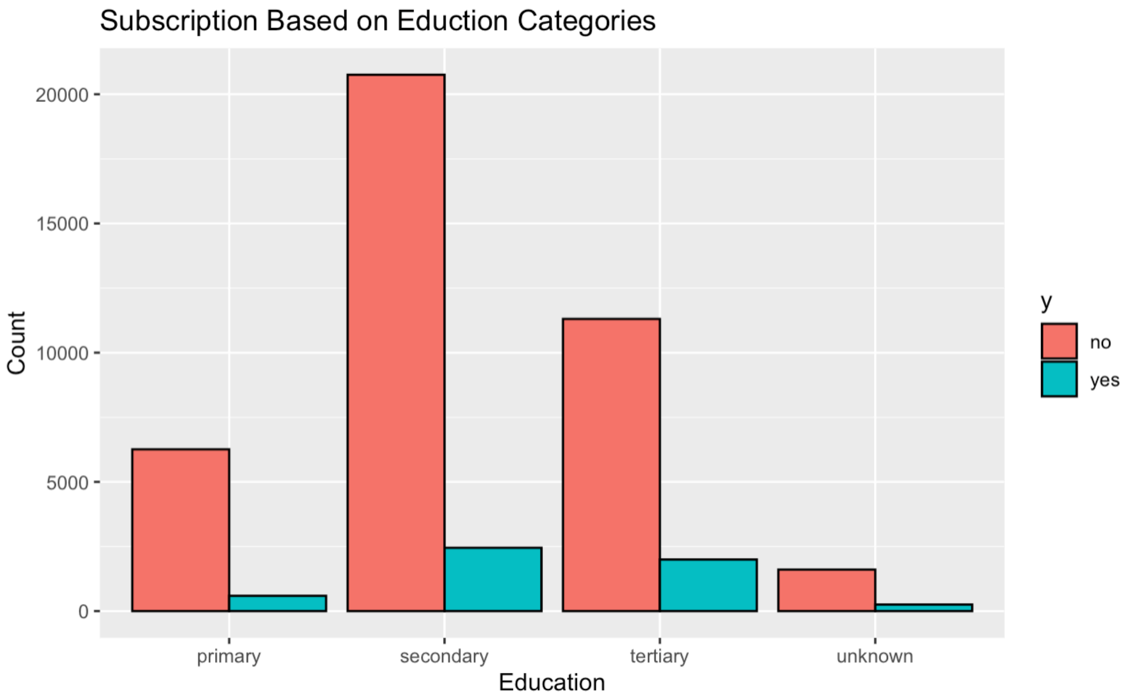
The density plot shows the bank balance for various categories of educated people.



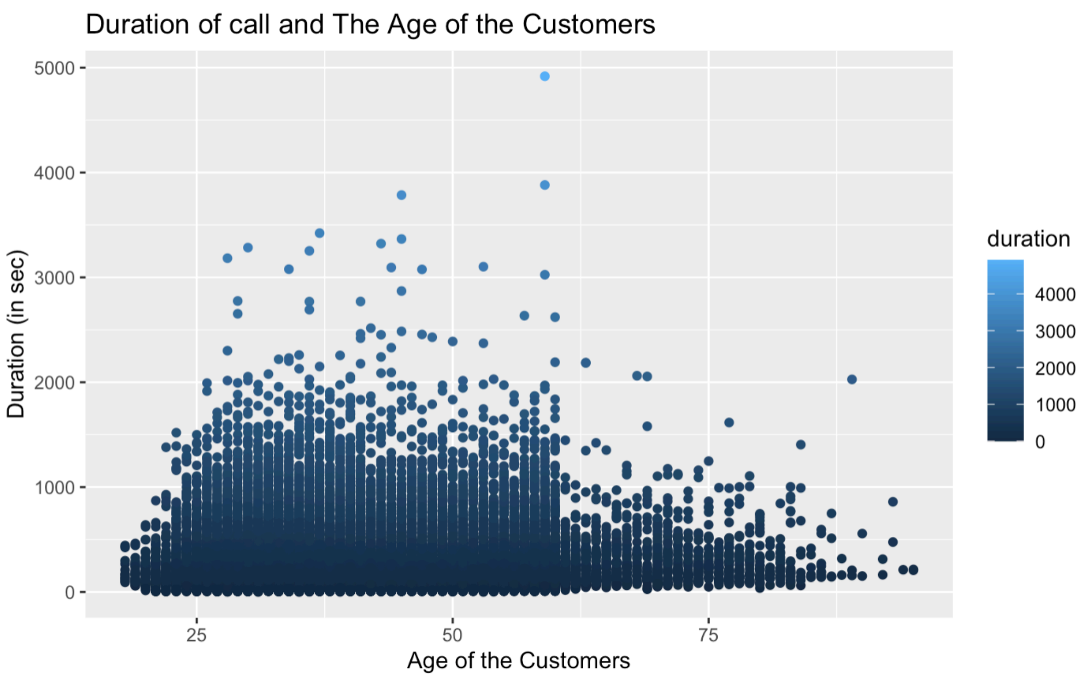
This plot shows the number of customers subscribing to term deposit according to their marital categories.



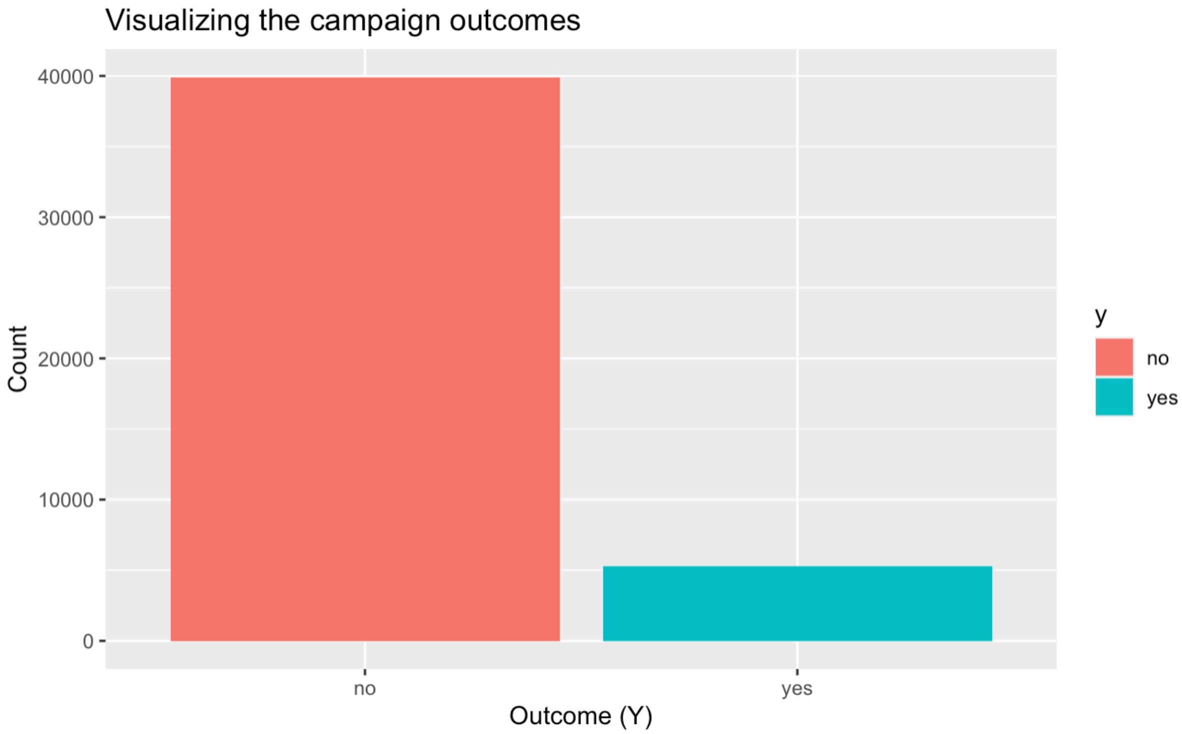
This is a boxplot shows the distribution of balance by marital status of the customers



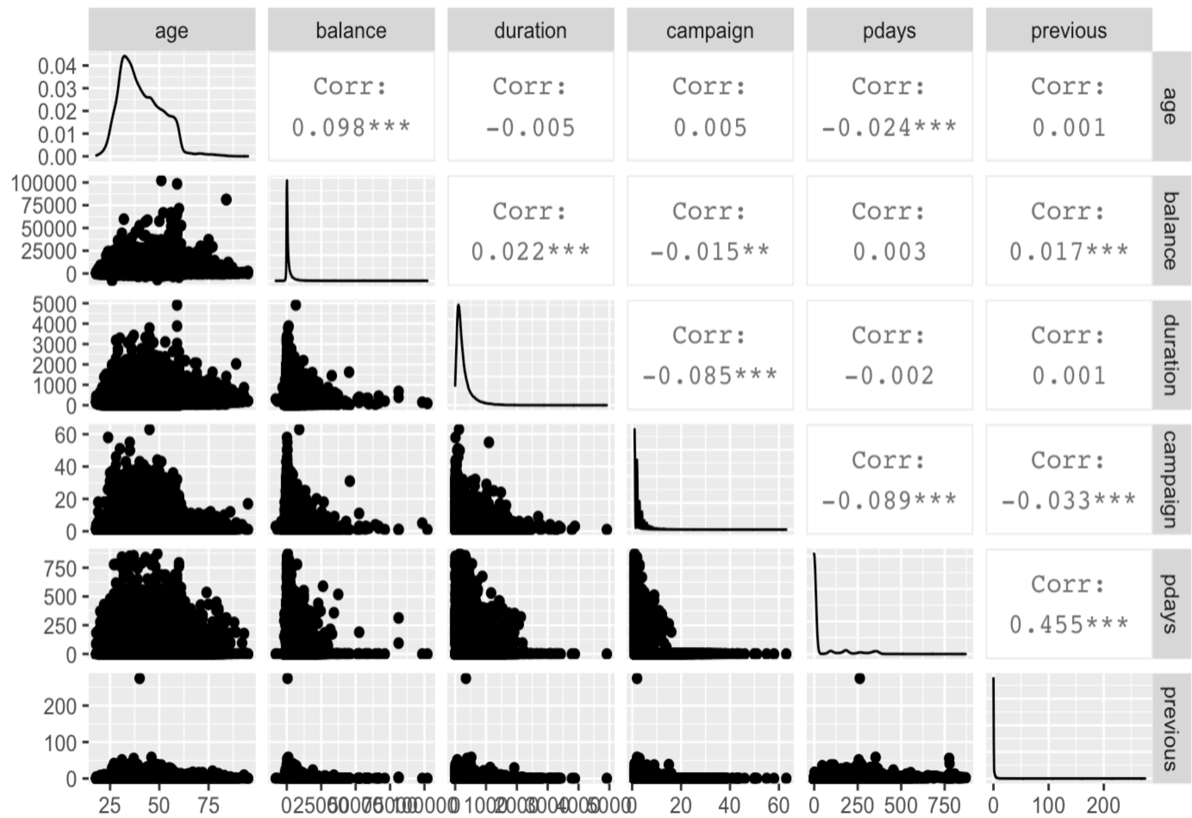
Similar to the above plot, this shows the number of customers subscribing to term deposit by their education categories.



The above plot shows the scatter plot for the duration of the call as per the age of the customers.



This shows the number of customers subscribing to the term deposit and the number of them not subscribing.



This plot shows the correlation of various numerically independent variables.

III. Data Preparation and Preprocessing:

Checking the data set for any missing records.

```
Checking the dataset for any missing variables.
```{r}
sum(is.na(bank_data))
```

 [1] 0
```

Converting the character variables into numeric type. The unknow responses of the variables are also taken into consideration by creating new variables.

```
bank_data$job_unknown <- ifelse(bank_data$job == "unknown", 1, 0)
bank_data$job <- as.numeric(as.factor(bank_data$job))
bank_data$marital <- as.numeric(as.factor(bank_data$marital))
bank_data$edu_unknown <- ifelse(bank_data$education == "unknown", 1, 0)
bank_data$education<- as.numeric(as.factor(bank_data$education))
bank_data$default <- ifelse(bank_data$default == "yes", 1, 0 )
bank_data$housing <- ifelse(bank_data$housing == "yes", 1, 0)
bank_data$loan <- ifelse(bank_data$loan == "yes", 1, 0)
bank_data$con_unknown <- ifelse(bank_data$contact == "unknown", 1, 0)
bank_data$contact <- as.numeric(as.factor(bank_data$contact))
bank_data$month <- as.numeric(as.factor(bank_data$month))
bank_data$pout_unknown <- ifelse(bank_data$poutcome == "unknown", 1, 0)
bank_data$poutcome <- as.numeric(as.factor(bank_data$poutcome))
bank_data$y <- ifelse(bank_data$y == "yes", 1, 0)
```

```
'data.frame':   45211 obs. of  21 variables:
 $ age         : int  58 44 33 47 33 35 28 42 58 43 ...
 $ job         : num  5 10 3 2 12 5 5 3 6 10 ...
 $ marital     : num  2 3 2 2 3 2 3 1 2 3 ...
 $ education   : num  3 2 2 4 4 3 3 3 1 2 ...
 $ default     : num  0 0 0 0 0 0 0 1 0 0 ...
 $ balance     : int  2143 29 2 1506 1 231 447 2 121 593 ...
 $ housing     : num  1 1 1 1 0 1 1 1 1 1 ...
 $ loan        : num  0 0 1 0 0 0 1 0 0 0 ...
 $ contact     : num  3 3 3 3 3 3 3 3 3 3 ...
 $ day         : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month       : num  9 9 9 9 9 9 9 9 9 9 ...
 $ duration    : int  261 151 76 92 198 139 217 380 50 55 ...
 $ campaign    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome    : num  4 4 4 4 4 4 4 4 4 4 ...
 $ y           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ job_unknown : num  0 0 0 0 1 0 0 0 0 0 ...
 $ edu_unknown : num  0 0 0 1 1 0 0 0 0 0 ...
 $ con_unknown : num  1 1 1 1 1 1 1 1 1 1 ...
 $ pout_unknown: num  1 1 1 1 1 1 1 1 1 1 ...
```

We now divide the data into training set (75%) and test data (25%) for all the models.

```
> #Splitting the data into Training & Test
> library(caTools)
> set.seed(100)
> split=sample.split(bank_data$y,SplitRatio = 0.75)
> bank_data_training = subset(bank_data, split==T)
> bank_data_test = subset(bank_data, split==F)
>
```

We normalize the data to maintain the balance and avoid one feature getting more weightage over the classification models.

```
> #Feature Scaling
> bank_data_training[,c(1,6,10,12,13:16)] = scale(bank_data_training[,c(1,6,10,12,13:16)])
> bank_data_test[,c(1,6,10,12,13:16)] = scale(bank_data_test[,c(1,6,10,12,13:16)] )
> head(bank_data_test)
          age job marital education default    balance housing loan contact
7  -1.2148690   5       3         3       0 -0.2900176       1    1       3
12 -1.1212857   1       3         2       0 -0.3075859       1    0       3
15  1.4990470   8       2         2       0 -0.3778594       1    0       3
24 -1.4956190   8       2         2       0 -0.4123797       1    0       3
27 -0.1854526   5       3         3       0 -0.3491952       1    0       3
28  1.0311304   3       2         2       0 -0.3929620       1    1       3
         day month   duration   campaign      pdays   previous  poutcome y
7  -1.299758     9 -0.1476692 -0.5745597 -0.4140792 -0.2998019 0.4482437 0
12 -1.299758     9 -0.4572521 -0.5745597 -0.4140792 -0.2998019 0.4482437 0
15 -1.299758     9 -0.3140700 -0.5745597 -0.4140792 -0.2998019 0.4482437 0
24 -1.299758     9  0.3360541 -0.5745597 -0.4140792 -0.2998019 0.4482437 0
27 -1.299758     9  0.1580439 -0.5745597 -0.4140792 -0.2998019 0.4482437 0
28 -1.299758     9 -0.4959500 -0.5745597 -0.4140792 -0.2998019 0.4482437 0
   job_unknown edu_unknown con_unknown pout_unknown
7            0           0           1            1
12           0           0           1            1
15           0           0           1            1
24           0           0           1            1
27           0           0           1            1
28           0           0           1            1
>
```

IV. Data Mining Techniques and Implementation

Logistic Regression:

```
Call:
glm(formula = y ~ ., family = binomial, data = bank_data_training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.7311  -0.4177  -0.2729  -0.1599   3.5620

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.066642   0.178137  -0.374  0.70832
age           0.047903   0.021674   2.210  0.02709 *
job           0.013601   0.006495   2.094  0.03626 *
marital       0.163416   0.036783   4.443 8.89e-06 ***
education     0.245456   0.032616   7.526 5.24e-14 ***
default      -0.186784   0.184914  -1.010  0.31244
balance       0.051344   0.016579   3.097  0.00196 **
housing      -0.893379   0.044897 -19.899  < 2e-16 ***
loan         -0.634239   0.067034  -9.461  < 2e-16 ***
contact      -0.083377   0.083512  -0.998  0.31809
day          -0.054188   0.020426  -2.653  0.00798 **
month         0.026085   0.006582   3.963 7.39e-05 ***
duration      1.036954   0.018439  56.236  < 2e-16 ***
campaign     -0.382305   0.036514 -10.470  < 2e-16 ***
pdays        -0.006029   0.033777  -0.178  0.85833
previous      0.005966   0.015048   0.396  0.69175
poutcome      1.069777   0.044959  23.795  < 2e-16 ***
job_unknown  -0.457733   0.269147  -1.701  0.08900 .
edu_unknown  -0.331548   0.113655  -2.917  0.00353 **
con_unknown  -1.095553   0.173810  -6.303 2.92e-10 ***
pout_unknown -3.382996   0.115894 -29.190  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24474  on 33908  degrees of freedom
Residual deviance: 17221  on 33888  degrees of freedom
AIC: 17263

Number of Fisher Scoring iterations: 6

   y_pred
       0    1
  0 9743  237
  1  900  422
```

```
[1] 0.8993983

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  11302

             | bank_data_test[, 17]
      y_pred |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |      9743 |       900 |     10643 |
             |     0.915 |     0.085 |     0.942 |
             |     0.976 |     0.681 |           |
             |     0.862 |     0.080 |           |
-------------|-----------|-----------|-----------|
           1 |       237 |       422 |       659 |
             |     0.360 |     0.640 |     0.058 |
             |     0.024 |     0.319 |           |
             |     0.021 |     0.037 |           |
-------------|-----------|-----------|-----------|
Column Total |      9980 |      1322 |     11302 |
             |     0.883 |     0.117 |           |
-------------|-----------|-----------|-----------|
```

Logistic regression is created on predictor variables in the model and we got an accuracy of 89.94%.

K-NN:

```
[1] 0.8765705
[1] 0.8748894
[1] 0.8913467
[1] 0.8896655
[1] 0.8922315
[1] 0.8938241
[1] 0.8952398
[1] 0.8950628
[1] 0.8950628
[1] 0.8947974

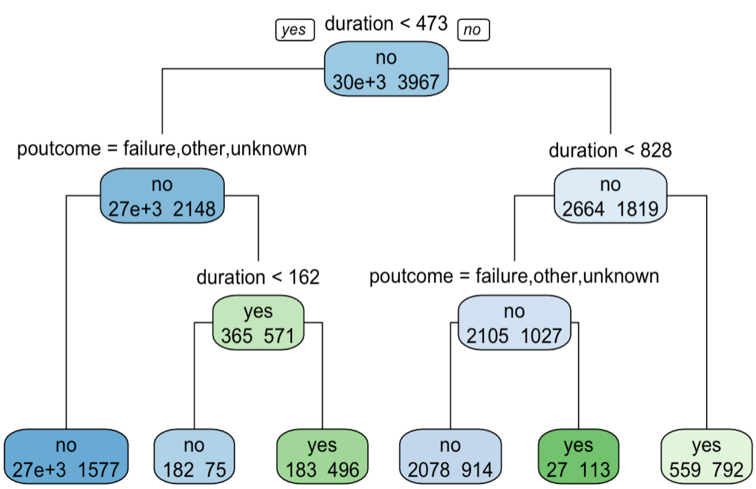

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  11302


             | bank_data_test[, 17]
  y_pred_kNN5 |           0 |           1 | Row Total |
-------------|-------------|-------------|-----------|
           0 |        9657 |         895 |     10552 |
             |       0.915 |       0.085 |     0.934 |
             |       0.968 |       0.677 |           |
             |       0.854 |       0.079 |           |
-------------|-------------|-------------|-----------|
           1 |         323 |         427 |       750 |
             |       0.431 |       0.569 |     0.066 |
             |       0.032 |       0.323 |           |
             |       0.029 |       0.038 |           |
-------------|-------------|-------------|-----------|
Column Total |        9980 |        1322 |     11302 |
             |       0.883 |       0.117 |           |
-------------|-------------|-------------|-----------|
> acc_KNN10<-sum(diag(ConMatKNN10))/sum(ConMatKNN10)
> acc_KNN10
[1] 0.8955937
```

When k = 5 we are getting the best accuracy of 89.56% accuracy. We can select k = 10 as the best value of k.

Decision Tree:




```
     y_CT2
        no  yes
  no  9730  250
  yes  877  445
[1] 0.9002831
```

Considering all the predictors we got accuracy of 90.03%, which is more than the logistic regression model.

Support Vector Machine (SVM):

```
Call:
svm(formula = y ~ ., data = bank_data_training, type = "C-classification", kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1

Number of Support Vectors:  7830

 ( 3923 3907 )


Number of Classes:  2

Levels:
 0 1



   y_SVM1
       0    1
  0 9839  141
  1  998  324
[1] 0.8992214
```

For SVM using the linear kernel we are getting 89.92% accuracy which is almost same as logistic regression.

Random Forest:

```
Call:
 randomForest(formula = y ~ ., data = RT_training, ntree = 500,      mtry = 4, nodesize = 5, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 9.3%
Confusion matrix:
       no  yes class.error
no  28948  994  0.03319752
yes  2158 1809  0.54398790
```

```
> CMRT= table(RT_test[,17],rf.pred)
> CMRT
      rf.pred
          no  yes
  no    9630  350
  yes    701  621
>
> acc_RT<-sum(diag(CMRT))/sum(CMRT)
> acc_RT
[1] 0.9070076
```

90.7% accuracy was obtained through Random Forest.

Naïve Bayes:

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = naba_data_training[-17], y = naba_data_training[,
    17])

A-priori probabilities:
naba_data_training[, 17]
        0         1
0.8830104 0.1169896

Conditional probabilities:
                        age
naba_data_training[, 17]     [,1]     [,2]
                       0 40.81668 10.15529
                       1 41.70885 13.44358

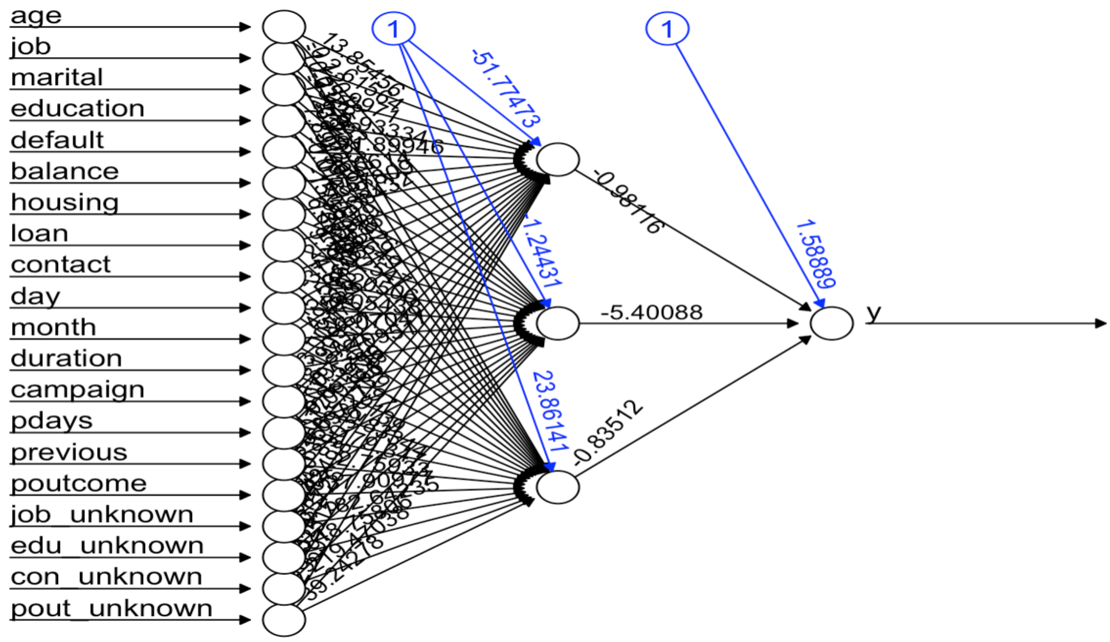
                        job
naba_data_training[, 17]      admin. blue-collar entrepreneur   housemaid  management     retired self-employed
                       0 0.113118696 0.229109612  0.034232850 0.027987442 0.202524881 0.043484069   0.035001002
                       1 0.118477439 0.129821023  0.022183010 0.018149735 0.249558861 0.096798588   0.036047391
                        job
naba_data_training[, 17]    services     student  technician  unemployed     unknown
                       0 0.094749850 0.016465166 0.169961926 0.027052301 0.006312204
                       1 0.072094782 0.049407613 0.164103857 0.037307789 0.006049912
```

```
> ConMatNB_train = table(naba_data_training[,17],naba_data_Pred_Training)
> ConMatNB_train
   naba_data_Pred_Training
        0     1
  0 27740  2202
  1  1808  2159
> ConMatNB = table(naba_data_test[,17],naba_data_Pred)
> ConMatNB
   naba_data_Pred
       0    1
  0 9225  755
  1  628  694
```

```
> acc_naba_data<-sum(diag(ConMatNB))/sum(ConMatNB)
> acc_naba_data
[1] 0.8776323
>
> #Accuracy train
> acc_naba_data_Train<-sum(diag(ConMatNB_train))/sum(ConMatNB_train)
> acc_naba_data_Train
[1] 0.8817423
```

The training accuracy is 88.17% and the model accuracy is 87.76% using the Naïve Bayes model and hence the least accurate among the other models.

Neural Nets:




```
> ConMAtNN = table(bank_data_test[,17],y_pred_NN)
> ConMAtNN
   y_pred_NN
       0    1
  0 9641  339
  1  787  535
> acc_NN<-sum(diag(ConMAtNN))/sum(ConMAtNN)
> acc_NN
[1] 0.9003716
```

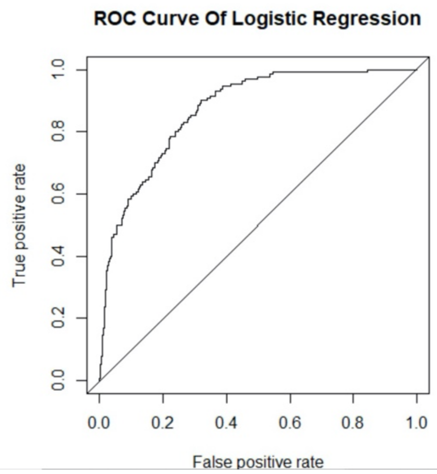
90.04% is achieved in Neural Nets.

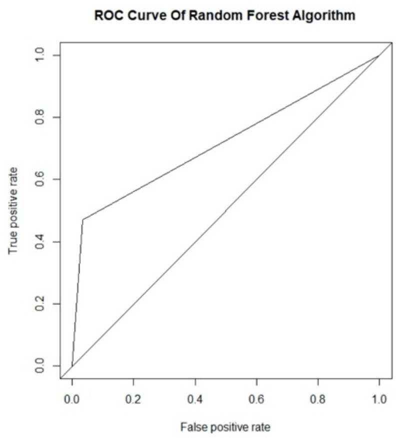
V. Performance Evaluation:

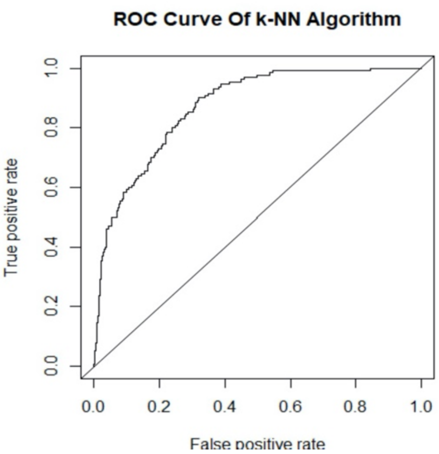
The efficiency of various models is listed below:

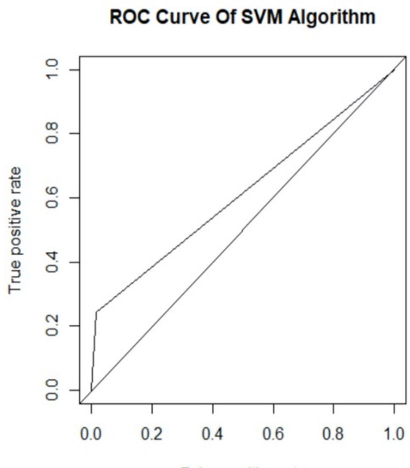
1) Logistic Regression, accuracy is 89.94%
2) K-NN, accuracy is 89.56%
3) Decision Tree, accuracy is 90.03%
4) Support Vector Machine, accuracy is 89.92%
5) Random Forest, accuracy is 90.7%
6) Naïve Bayes, accuracy is 88.17%
7) Neural Nets, accuracy is 90.04%

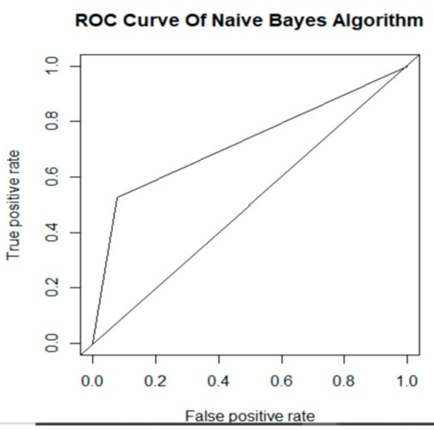
ROC Curves for different Models:

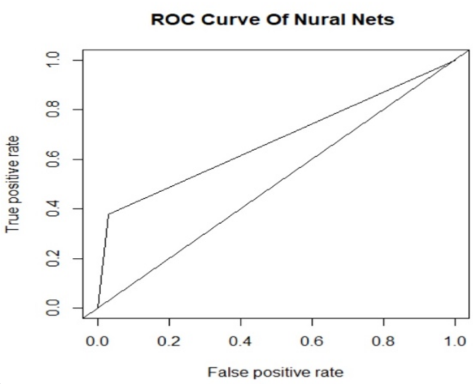












VI. Conculsion

Case study was conducted on the Portuguese bank to improve their marketing campaign using the machine learning algorithms. The results obtained for various models were extremely good as most of the attributes contributed in the direction of the model. Considering duration of call in seconds as the main attribute, Logistic regression and Random Forest as among the best models for the study.

The bank can improve their marketing campaign by targeting the crowd which is more likely subscribe their product rather than investing towards the crowd who are unlikely to buy it. The target crowd can be identified by using the models mentioned above, thus resulting in an effective and more efficient marketing campaign.

VII. Summary

Steps involved in the case study are as follows:

1) Data selection: The dataset was found in the UCI repository
2) Data Exploration: Primary analysis of the dataset
3) Data Cleaning: Involves removing null and missing records, Normalizing the data
4) Data Visualization: Bar-chat, Density Graph, Correlation Matrix, Boxplot, Scatter Plot
5) Splitting the data into Training set 75% and Test set 25%
6) Model Building: Logistic regression, K-NN, Decision Tree, Support Vector Machine, Random Forest, Naïve Bayes and Neural Nets
7) Performance Evaluation: Confusion Matrix and ROC Curves
8) Conclusion: Selecting Logistic Regression as the best model based on accuracy and ROC Curve.

Appendix: R Code for Case Study

```
#install.packages("ggplot2")
#install.packages("gmodels")
#install.packages("ROCR")
#install.packages("caTools")
#install.packages("rpart")
#install.packages("rpart.plot")
#install.packages("e1071")
#install.packages("randomForest")
#install.packages("neuralnet")

library(GGally)
library(ggplot2)
library(gmodels)
library(ROCR)
library(caTools)
library(rpart)
library(rpart.plot)
library(e1071)
library(randomForest)
library(neuralnet)

getwd()
setwd(~/Downloads)
# Importing the dataset
bank_data <- read.csv("~/Downloads/bank (1)/bank-full.csv", sep=";",
stringsAsFactors = FALSE)
exp_bankdata <- read.csv("~/Downloads/bank (1)/bank-full.csv", sep = ";",
stringsAsFactors = TRUE)
head(bank_data,5)

#bank_data <- read.csv("C:/Users/Amit/Desktop/Reema/Project/bank (1)/bank-
full.csv", sep=";", stringsAsFactors = FALSE)
#exp_bankdata <- read.csv("C:/Users/Amit/Desktop/Reema/Project/bank
(1)/bank-full.csv", sep = ";", stringsAsFactors = TRUE)
#head(bank_data,5)

#Checking the dimensions of the dataset
dim(bank_data)

str(bank_data)
str(exp_bankdata)
```

```
summary(bank_data)

#Listing out the various Categories
levels(exp_bankdata$job)
levels(exp_bankdata$marital)
levels(exp_bankdata$education)
levels(exp_bankdata$default)
levels(exp_bankdata$housing)
levels(exp_bankdata$loan)
levels(exp_bankdata$contact)
levels(exp_bankdata$y)

#Visualizing the Dataset
library(ggplot2)
ggplot(bank_data, aes(x = job, y = balance, fill = month)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 45)) + xlab("Job Categories") + ylab("Balance") + ggtitle("Barplot for different Job Categories and Balance in Account")

ggplot(bank_data) + geom_density(aes(x = log(balance), fill = education), alpha = 0.4) + xlab("Balance") + ylab("Density") + ggtitle("Density Plot for the Education and the Bank Balance")


ggplot(bank_data, aes(x = marital, fill = y)) + geom_bar(position = "dodge", colour = "black") + xlab("Marital Status") + ylab("Count") + ggtitle("Subscription based on Marital Status")

ggplot(bank_data, aes(x = education, fill = y)) + geom_bar(position = "dodge", colour = "black") + xlab("Education") + ylab("Count")+ggtitle("Subscription Based on Eduction Categories")

ggplot(bank_data, aes(x = marital, y = balance, colour = marital)) + geom_boxplot(size = 1) + ylim(-1000, 10000) + xlab("Marital Status") + ylab("Balance") + ggtitle("Boxplot of Marital Status and the Balance in Account") + theme(legend.position = "none")

ggplot(bank_data, aes(x = age, y = duration, colour = duration)) + geom_point() + xlab("Age of the Customers") + ylab("Duration (in sec)") + ggtitle("Duration of call and The Age of the Customers")

library(GGally)
```

```
ggpairs(exp_bankdata[, c(1, 6, 12, 13, 14, 15)])

ggplot(bank_data, aes(x = y, fill = y)) + geom_bar() + xlab("Outcome (Y)") +
ylab("Count") + ggtitle("Visualizing the campaign outcomes")


#Data Cleaning and Preprocessing
#Checking the dataset for any missing variables.
sum(is.na(bank_data))

#Converting the character variables to numeric type. The "unknown" responses
are also considered by creating new variables.
bank_data$job_unknown <- ifelse(bank_data$job == "unknown", 1, 0)
bank_data$job <- as.numeric(as.factor(bank_data$job))
bank_data$marital <- as.numeric(as.factor(bank_data$marital))
bank_data$edu_unknown <- ifelse(bank_data$education == "unknown", 1, 0)
bank_data$education<- as.numeric(as.factor(bank_data$education))
bank_data$default <- ifelse(bank_data$default == "yes", 1, 0 )
bank_data$housing <- ifelse(bank_data$housing == "yes", 1, 0)
bank_data$loan <- ifelse(bank_data$loan == "yes", 1, 0)
bank_data$con_unknown <- ifelse(bank_data$contact == "unknown", 1, 0)
bank_data$contact <- as.numeric(as.factor(bank_data$contact))
bank_data$month <- as.numeric(as.factor(bank_data$month))
bank_data$pout_unknown <- ifelse(bank_data$poutcome == "unknown", 1, 0)
bank_data$poutcome <- as.numeric(as.factor(bank_data$poutcome))
bank_data$y <- ifelse(bank_data$y == "yes", 1, 0)

#Checking the Structure of the bank dataset
str(bank_data)

#Splitting the data into Training & Test
library(caTools)
set.seed(100)
split=sample.split(bank_data$y,SplitRatio = 0.75)
bank_data_training = subset(bank_data, split==T)
bank_data_test = subset(bank_data, split==F)


#Feature Scaling
bank_data_training[,c(1,6,10,12,13:16)] =
scale(bank_data_training[,c(1,6,10,12,13:16)])
bank_data_test[,c(1,6,10,12,13:16)] = scale(bank_data_test[,c(1,6,10,12,13:16)]
)
```

```
head(bank_data_test)

#Data Mining Techniques and Implementation
#Logistic Regression

classifier = glm(formula = y~.,
             family =binomial,
             data= bank_data_training)
summary(classifier)
Prob_pred = predict(classifier, type= 'response',
               newdata= bank_data_test[-17])

#Confusion Matrix
y_pred= ifelse(Prob_pred>0.5,1,0)
ConMatLog = table(bank_data_test[,17],y_pred)

ConMatLog

#Accuracy
acc_Log1<-sum(diag(ConMatLog))/sum(ConMatLog)
acc_Log1
library(gmodels)
CrossTable(y_pred,bank_data_test[,17],prop.chisq = F)

library(ROCR)
ROCRpredLogi <- prediction(Prob_pred,bank_data_test$y)
ROCRperfLogi <- performance(ROCRpredLogi, 'tpr','fpr')
plot(ROCRperfLogi, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of Logistic Regression")

#KNN
library(class)
#Fitting K-NN to the training Dataset and Predicting  the Test results
#Accuracy when k=1
y_pred_kNN1 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 1)
ConMatKNN1 = table(bank_data_test[,17],y_pred_kNN1)
acc_KNN1<-sum(diag(ConMatKNN1))/sum(ConMatKNN1)
acc_KNN1

#Accuracy when k=2
```

```
y_pred_kNN2 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 2)
ConMatKNN2 = table(bank_data_test[,17],y_pred_kNN2)
acc_KNN2<-sum(diag(ConMatKNN2))/sum(ConMatKNN2)
acc_KNN2

#Accuracy when k=3
y_pred_kNN3 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 3)
ConMatKNN3 = table(bank_data_test[,17],y_pred_kNN3)
acc_KNN3<-sum(diag(ConMatKNN3))/sum(ConMatKNN3)
acc_KNN3

#Accuracy when k=4
y_pred_kNN4 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 4)
ConMatKNN4 = table(bank_data_test[,17],y_pred_kNN4)
acc_KNN4<-sum(diag(ConMatKNN4))/sum(ConMatKNN4)
acc_KNN4

#Accuracy when k=5
y_pred_kNN5 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 5)
ConMatKNN5 = table(bank_data_test[,17],y_pred_kNN5)
acc_KNN5<-sum(diag(ConMatKNN5))/sum(ConMatKNN5)
acc_KNN5

#Accuracy when k=6
y_pred_kNN6 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 6)
ConMatKNN6 = table(bank_data_test[,17],y_pred_kNN6)
acc_KNN6<-sum(diag(ConMatKNN6))/sum(ConMatKNN6)
acc_KNN6

#Accuracy when k=7
y_pred_kNN7 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 7)
```

```
ConMatKNN7 = table(bank_data_test[,17],y_pred_kNN7)
acc_KNN7<-sum(diag(ConMatKNN7))/sum(ConMatKNN7)
acc_KNN7

#Accuracy when k=8
y_pred_kNN8 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 8)
ConMatKNN8 = table(bank_data_test[,17],y_pred_kNN8)
acc_KNN8<-sum(diag(ConMatKNN8))/sum(ConMatKNN8)
acc_KNN8

#Accuracy when k=9
y_pred_kNN9 = knn(train = bank_data_training[,-17],
              test= bank_data_test[,-17],
              cl=bank_data_training[,17],k= 9)
ConMatKNN9 = table(bank_data_test[,17],y_pred_kNN9)
acc_KNN9<-sum(diag(ConMatKNN9))/sum(ConMatKNN9)
acc_KNN9

#Accuracy when k=10
y_pred_kNN10 = knn(train = bank_data_training[,-17],
               test= bank_data_test[,-17],
               cl=bank_data_training[,17],k= 10)
ConMatKNN10 = table(bank_data_test[,17],y_pred_kNN10)
acc_KNN10<-sum(diag(ConMatKNN10))/sum(ConMatKNN10)
acc_KNN10

CrossTable(y_pred_kNN5,bank_data_test[,17],prop.chisq = F)

#ROC knn
ROCRpredknn <- prediction(as.numeric(y_pred_kNN10),bank_data_test$y)
ROCRperfknn <- performance(ROCRpredknn, 'tpr','fpr')
plot(ROCRperfknn, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of k-NN Algorithm")

#Decision Tree
library(caTools)
set.seed(100)
split=sample.split(exp_bankdata$y,SplitRatio = 0.75)
CTdata2_training = subset(exp_bankdata, split==T)
CTdata2_test = subset(exp_bankdata, split==F)
```

```
library(rpart )
library(rpart.plot)
CT_model2<- rpart(formula = y~.,
               data =CTdata2_training, method='class',maxdepth=5)
print(CT_model2)
rpart.plot(CT_model2,type = 1, extra = 1, split.font = 1, varlen = -10)
summary(CT_model2)
y_CT2<-predict(CT_model2,newdata= CTdata2_test[,-17], type ='class')

CMCT2= table(CTdata2_test[,17],y_CT2)
CMCT2

acc_CT2<-sum(diag(CMCT2))/sum(CMCT2)
acc_CT2
library(gmodels)
CrossTable(y_CT2,CTdata2_test[,17],prop.chisq = F)

#ROC CT
ROCRpredCT1 <- prediction(as.numeric(y_CT2),bank_data_test$y)
ROCRperfCT1<- performance(ROCRpredCT1, 'tpr','fpr')
plot(ROCRperfCT1, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of Decision Tree ")

#Support Vector Machines
library(e1071)
SVM_Model1 = svm(formula = y~.,
              data= bank_data_training,
              type='C-classification',
              kernel='linear')
summary(SVM_Model1)
y_SVM1 = predict(SVM_Model1,newdata= bank_data_test[-17])

#Creating the Matrix and finding accuracy
ConMatSVM1 = table(bank_data_test[,17],y_SVM1)
ConMatSVM1
acc_SVM<-sum(diag(ConMatSVM1))/sum(ConMatSVM1)
acc_SVM

#ROC Support Vector Machine
ROCRpredSVM <- prediction(as.numeric(y_SVM1),bank_data_test$y)
ROCRperfSVM <- performance(ROCRpredSVM, 'tpr','fpr')
```

```
plot(ROCRperfSVM, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of SVM Algorithm")

#Random Forest
library(caTools)
set.seed(100)
split=sample.split(exp_bankdata$y,SplitRatio = 0.75)
RT_training = subset(exp_bankdata, split==T)
RT_test = subset(exp_bankdata, split==F)

library(randomForest)
RanFor <- randomForest(y ~ ., data = RT_training, ntree = 500,
                mtry = 4, nodesize = 5, importance = TRUE)
rf.pred <- predict(RanFor, RT_test[,-17])

#confusion Matrix
CMRT= table(RT_test[,17],rf.pred)
CMRT

acc_RT<-sum(diag(CMRT))/sum(CMRT)
acc_RT

ROCRpredRF <- prediction(as.numeric(rf.pred),RT_test[,17] )
ROCRperfRF <- performance(ROCRpredRF, 'tpr','fpr')
plot(ROCRperfRF, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of Random Forest Algorithm")


#Naive Bayes
library(e1071)
naba_data  <-  read.csv("C:/Users/Amit/Desktop/Reema/Project/bank  (1)/bank-
full.csv", sep=";", stringsAsFactors = FALSE)
naba_data$y<- factor(bank_data$y, levels=c(0,1))

#Spliting data into Training and Test
library(caTools)
set.seed(100)
split=sample.split(naba_data$y,SplitRatio = 0.75)
naba_data_training = subset(naba_data, split==T)
naba_data_test = subset(naba_data, split==F)
naba_data1 =naiveBayes(x=naba_data_training[-17],
```

```
            y= naba_data_training[,17])

#predicting the Training results
naba_data_Pred_Training= predict(naba_data1,newdata= naba_data_training[,-
17])
naba_data_Pred_Training

#predicting the Test results
naba_data_Pred = predict(naba_data1,newdata= naba_data_test[,-17])
naba_data_Pred

#Creating Confusion matrix
ConMatNB_train = table(naba_data_training[,17],naba_data_Pred_Training)
ConMatNB_train
ROCRpredRF <- prediction(as.numeric(rf.pred),RT_test[,17] )
ROCRperfRF <- performance(ROCRpredRF, 'tpr','fpr')
plot(ROCRperfRF, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of Random Forest Algorithm")

ConMatNB = table(naba_data_test[,17],naba_data_Pred)
ConMatNB

#Accuracy test
acc_naba_data<-sum(diag(ConMatNB))/sum(ConMatNB)
acc_naba_data

#Accuracy train
acc_naba_data_Train<-sum(diag(ConMatNB_train))/sum(ConMatNB_train)
acc_naba_data_Train

#ROC for Naive Bayes
ROCRprednaba_data                                              <-
prediction(as.numeric(naba_data_Pred),naba_data_test[,17] )
ROCRperfnaba_data <- performance(ROCRprednaba_data, 'tpr','fpr')
plot(ROCRperfnaba_data, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of Naive Bayes Algorithm")

#Neural Networks
library(neuralnet)
neunet<- neuralnet(y~.,
            data = bank_data_training,
```

```
              linear.output = F,
              hidden = 3)
plot(neunet, rep="best")
predictnn <- neuralnet::compute(neunet,bank_data_test[,-17])
pre_nn <- predict(neunet,newdata= bank_data_test[,-17])
y_pred_NN= ifelse(pre_nn>0.5,1,0)

#confusion Matrix and accuracy
ConMAtNN = table(bank_data_test[,17],y_pred_NN)
ConMAtNN
acc_NN<-sum(diag(ConMAtNN))/sum(ConMAtNN)
acc_NN

#ROC Neural Networks
ROCRpredNN <- prediction(as.numeric(y_pred_NN),bank_data_test[,17] )
ROCRperfNN <- performance(ROCRpredNN, 'tpr','fpr')
plot(ROCRperfNN, colorize = F, text.adj = c(-0.2,1.7))
abline(a=0,b=1)
title(main="ROC Curve Of Neural Networks ")
```