BANK MARKETING DATASET ANALYSIS

Name : Anh Thi Nguyet Nguyen

NetID : xc8374

Class : Stat6620 _ Session 2

1. INTRODUCTION

In the age of Big-Data, data-driven techniques are applied in every aspect of a business and will help businesses scale and have proper solutions. The bank in this project's dataset is considering how to optimize its campaign in future. Making a data-driven decision is very essential to suggest the marketing manager about effective client selection, which would increase the conversion rate.

My main task in this project will be analyzing the dataset to find out how customer' banking habits' characteristics will result in bank's term deposit subscription. The model will allow the bank to predict whether a particular client will subscribe to term deposit or not. If classifier has very high accuracy it can help the manager to filter clients and use available resources more efficiently to achieve the campaign goal. A good model will support the bank in future marketing strategies. In this analysis, varied data analysis methods - supervised and unsupervised- such as: decision tree, random forest, k-means clustering will be applied. The 5-steps procedure will also be used as a main structure for the analysis.

2. DATA ANALYSIS

2.1 Collecting data

The dataset is downloaded directly from the UCI Machine Learning Repository website, below is its name and the URL for downloading Name : Bank Marketing Source : http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

There are four datasets: 1) bank-additional-full.csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014] 2) bank-additional.csv with 10% of the examples (4119), randomly selected from 1), and 20 inputs. 3) bank-full.csv with all examples and 17 inputs, ordered by date (older version of this dataset with less inputs). 4) bank.csv with 10% of the examples and 17 inputs, randomly selected from 3 (older version of this dataset with less inputs).

In this analysis, I will use the bank-additional-full.csv file

The dataset has 41188 records with 21 variables with definition below :

age - Age of the client- (numeric)

job - Client's occupation - (categorical) (admin, bluecollar, entrepreneur, housemaid, management, retired, selfemployed, services, student, technician, unemployed, unknown)

marital - Client's marital status - (categorical) (divorced, married, single, unknown, note: divorced means divorced or widowed)

education - Client's education level - (categorical) (basic.4y, basic.6y, basic.9y, high.school, illiterate, professional.course, university.degree, unknown)

default - Indicates if the client has credit in default - (categorical) (no, yes, unknown)

housing - Does the client as a housing loan? - (categorical) (no, yes, unknown)

loan - Does the client as a personal loan? - (categorical) (no, yes, unknown')

contact - Type of communication contact - (categorical) (cellular, telephone)

month - Month of last contact with client - (categorical) (January - December)

day_of_week - Day of last contact with client - (categorical) (Monday - Friday)

duration - Duration of last contact with client, in seconds - (numeric) For benchmark purposes only, and not reliable for predictive modeling

campaign - Number of client contacts during this campaign - (numeric) (includes last contact)

pdays - Number of days from last contacted from a previous campaign - (numeric) (999 means client was not previously contacted)

previous - Number of client contacts performed before this campaign - (numeric)

poutcome - Previous marketing campaign outcome - (categorical) (failure, nonexistent , success)

emp.var.rate - Quarterly employment variation rate - (numeric)

cons.price.idx - Monthly consumer price index - (numeric)

cons.conf.idx - Monthly consumer confidence index - (numeric)

euribor3m - Daily euribor 3 month rate - (numeric)

nr.employed - Quarterly number of employees - (numeric)

Output variable (desired target) - Term Deposit - subscription verified (binary: 'yes','no')

2.2 Exploring and preparing the data

We will use the read.csv() function to load the data into R and examine the structure of the dataset by the str() command.

```
bank_marketing <- read.csv("bank-additional-full.csv", header = TRUE, sep= ";
")
str(bank_marketing)

## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1
8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2
2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4
3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1
1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3
3 ...
##  $ loan          : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1
1 ...
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2
2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7
7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2
2 2 2 ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2
2 2 2 2 2 2 ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -3
6.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

As could be seen, the dataset contains 41188 obs. of 21 variables with 10 numeric and 21 factor variables. Our target response here is the "y", which is described as a factor with 2 levels "no","yes". Missing data are common occurrence and can have a significant effect on the conclusions that can be drawn from the data. So as a part of data cleansing process, we also need to check if the dataset has any missing values and remove those missing entries if needed. Fortunately, this dataset has no missing values at all so we could move on exploring the dataset by some visualization command.

```
length(which(is.na(bank_marketing)))
na_rm <- na.omit(bank_marketing)
dim(bank_marketing)
dim(na_rm)
```

Next, we will have a look at some of the customers' features using the summary () command. The Median age of the customers is 38, administrative jobs are the most popular jobs and university degree is the most common educational feature among them.

2.3 Training a model on the dataset and Evaluating models' performance

2.3.1 Decision Tree

As usual, our first step in training a model on the dataset is to split the dataset in to training and testing sub-sets randomly. The dataset is partitioned into 2 parts: 70% is training and the rest is testing set.

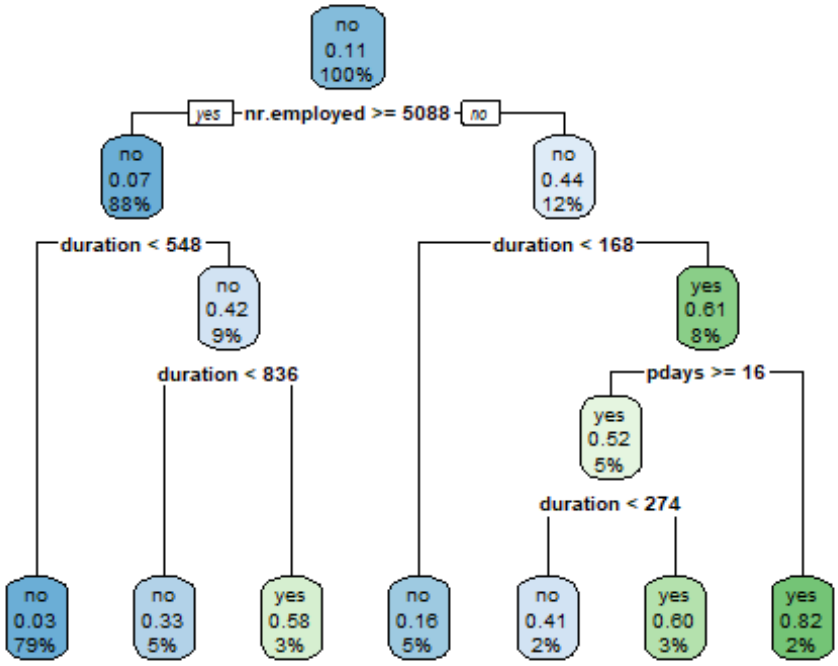
```
# create training and testing data
library(rpart)
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

set.seed(2000)
train_idx <- createDataPartition(bank_marketing$y, p =0.7, list = FALSE, time
s = 1)
```

```
train_data <- bank_marketing[train_idx,]
test_data <- bank_marketing[-train_idx,]

# check
dim(train_data)

## [1] 28832    21

dim(test_data)

## [1] 12356    21

table(train_data$y)

##
##    no   yes
## 25584  3248

table(test_data$y)

##
##    no   yes
## 10964  1392
```

Now we used the training dataset to train a decision tree model using rpart package.

```
library(rpart.plot)
library(entropy)
library(C50)
# Traing model using rpart
model_rpart <- rpart(y ~ ., data = train_data, method = "class")
model_rpart

## n= 28832
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 28832 3248 no (0.88734739 0.11265261)
##    2) nr.employed>=5087.65 25312 1691 no (0.93319374 0.06680626)
##      4) duration< 547.5 22787  624 no (0.97261597 0.02738403) *
##      5) duration>=547.5 2525 1067 no (0.57742574 0.42257426)
##       10) duration< 835.5 1573  514 no (0.67323586 0.32676414) *
##       11) duration>=835.5 952  399 yes (0.41911765 0.58088235) *
##    3) nr.employed< 5087.65 3520 1557 no (0.55767045 0.44232955)
##      6) duration< 168.5 1308  214 no (0.83639144 0.16360856) *
##      7) duration>=168.5 2212  869 yes (0.39285714 0.60714286)
##       14) pdays>=15.5 1565  755 yes (0.48242812 0.51757188)
##         28) duration< 274.5 646  263 no (0.59287926 0.40712074) *
##         29) duration>=274.5 919  372 yes (0.40478781 0.59521219) *
##       15) pdays< 15.5 647  114 yes (0.17619784 0.82380216) *
```

As we can see, all 28,832 individuals of the training set begin at the root node, of which 25312 have quarterly number of employees >=5087.65. Because quarterly number of employees was used first in the tree, it is the single most important predictor of our target response. Nodes indicated by * are terminal or leaf nodes, which means that they result in a prediction. For example, node 6 has a duration of 168.5. When the tree is used for predictions, any samples with nr.employed < 5087.65 and duration < 168.5 would therefore be predicted to have a "no" response.A visual view of the tree could help us have a better insight into the decision model :

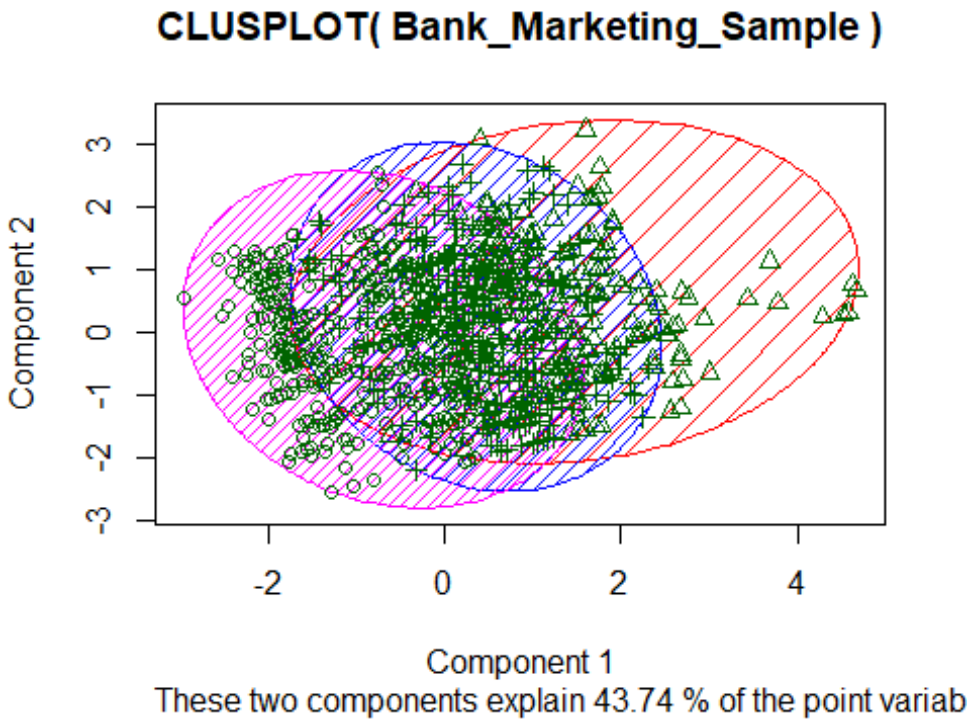


2.3.2 k-means Clustering

Now suppose the bank is trying to segment its customers into group to develop appropriate marketing strategies, I will use the k-means clustering method to cluster customer into subgroups. I took a sample dataset of 1000 records to cluster. The 2 tables below represent the centers of the 3 clusters and the size of each cluster.

I want to cluster the customers into 3 groups with the k-means algorithm, setting the k= 3, we could see above the centers of each element in each group. Generally, group 1 could be described as customers in teenage with marital status most likely to be "single" and the other 2 groups are more likely to be in their adulthood with marital status is "married". The customers tend to represent cluster 1 the most as the size for 3 clusters are: 430, 328, 242 accordingly. Researching each group's features thoroughly could help the bank develop helpful marketing campaign.

```
##        Age      Job  Marital Education  Housing     Loan
## 1 15.54176 4.841542 2.400428  5.017131 2.079229 1.291221
## 2 41.63402 4.819588 1.804124  4.298969 2.278351 1.309278
## 3 27.51622 4.654867 1.985251  4.522124 2.041298 1.297935

Bank_Marketing_k3$size

## [1] 467 194 339

library(cluster)
clusplot(Bank_Marketing_Sample, Bank_Marketing_k3$cluster, color=TRUE, shade=
TRUE, lines=0)
```



2.4. Evaluating the decision tree's performance

Upon running the commands to apply the decision tree to make predictions on the test dataset, We could generate the confusion matrix to evaluate the accuracy of the decision tree. As we could see from the results illustrated below, there were 1083 cases out of 12356 were misclassified, making the accuracy rate of the model stand at 91.24%.

```
predicted <- predict(model_rpart, test_data, type = "class")
table(predicted)
```

```
## predicted
##    no   yes
## 11287  1069
```

```
confusion.matrix <- prop.table(table(predicted, test_data$y))
confusion.matrix
```

```
##
## predicted         no        yes
##       no  0.85658789 0.05689544
##       yes 0.03075429 0.05576238
```

```
confusionMatrix(predicted,test_data$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  10584   703
##        yes   380   689
##
##                Accuracy : 0.9124
##                  95% CI : (0.9072, 0.9173)
##     No Information Rate : 0.8873
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5122
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9653
##             Specificity : 0.4950
##          Pos Pred Value : 0.9377
##          Neg Pred Value : 0.6445
##              Prevalence : 0.8873
##          Detection Rate : 0.8566
##    Detection Prevalence : 0.9135
##       Balanced Accuracy : 0.7302
##
##        'Positive' Class : no
##
```

91.24% for the accuracy of the decision tree model is not a bad result. However, as decision tree is not really a strong tool for prediction. I will try to improve the model using the random forest algorithm later to see if there's any improvement for the model. But next, let's just try to cluster the dataset.

2.5 : Improving model's performance

I will now try to build a Random Forest model on the Bank Marketing dataset to see if I can improve the performance of the decision tree as Random forest is normally considered a more advanced classification model than a simple decision tree. I use the randomForest package with the support of the caret package to test the performance of the model.

```
confusionMatrix(predicted_rf,test_data$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  10549   673
##        yes   415   719
##
##                Accuracy : 0.9119
##                  95% CI : (0.9068, 0.9169)
##     No Information Rate : 0.8873
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5208
##  Mcnemar's Test P-Value : 6.624e-15
##
##             Sensitivity : 0.9621
##             Specificity : 0.5165
##          Pos Pred Value : 0.9400
##          Neg Pred Value : 0.6340
##              Prevalence : 0.8873
##          Detection Rate : 0.8538
##    Detection Prevalence : 0.9082
##       Balanced Accuracy : 0.7393
##
##        'Positive' Class : no
##
```

Running random Forest in the same process for the training and testing dataset and use cross validation method to evaluate the model performance. We could see the final result shows that the accuracy for Random Forest model is 91.34 % , which is no improvement from the dicision tree above.

3 Conclusion :

The Dicision Tree model could perform quite well on the Bank Marketing dataset and can help the bank draw some decision. The Random Forest model was applied but showed no improvement in accuracy. Logistic regression, ANN and SVM may be appropriate for dataset hence can be applied to check for better accuracy. As outcome has only two values Logistic regression may seem appropriate choice but as there are more categorical

variables Logistic regression may not result in better accuracy, I will try Neural Networks in the future for the dataset.

Appendix

Belows are commands used in this project

```
bank_marketing <- read.csv("bank-additional-full.csv", header = TRUE, sep= ";
")
str(bank_marketing)


length(which(is.na(bank_marketing)))
na_rm <- na.omit(bank_marketing)
dim(bank_marketing)
dim(na_rm)


summary(bank_marketing)


hist(bank_marketing$age, breaks=30, prob = TRUE, xlab = "Age", main = "Histog
ram of Age")
lines(density(bank_marketing$age), col = "blue")



library(rpart)
library(caret)
set.seed(2000)
train_idx <- createDataPartition(bank_marketing$y, p =0.7, list = FALSE, time
s = 1)
train_data <- bank_marketing[train_idx,]
test_data <- bank_marketing[-train_idx,]

# check
dim(train_data)
dim(test_data)

table(train_data$y)
table(test_data$y)


library(rpart.plot)
library(entropy)
library(C50)
# Traing model using rpart
model_rpart <- rpart(y ~ ., data = train_data, method = "class")
```

```r
model_rpart


summary(model_rpart)
rpart.plot(model_rpart)



Bank_Marketing2 <- data.frame(as.numeric(as.factor(bank_marketing$age)),
                              as.numeric(as.factor(bank_marketing$job)),
                              as.numeric(as.factor(bank_marketing$marital))
,
                              as.numeric(as.factor(bank_marketing$education
)),
                              as.numeric(as.factor(bank_marketing$housing))
,
                              as.numeric(as.factor(bank_marketing$loan)))


colnames(Bank_Marketing2) <- c("Age", "Job", "Marital", "Education", "Housing
", "Loan")

Bank_Marketing_Sample <- Bank_Marketing2[sample(nrow(Bank_Marketing2),1000),]


set.seed(12345)
Bank_Marketing_k3 <- kmeans(Bank_Marketing_Sample, centers=3)

Bank_Marketing_k3$centers
Bank_Marketing_k3$size

library(cluster)
clusplot(Bank_Marketing_Sample, Bank_Marketing_k3$cluster, color=TRUE, shade=
TRUE, lines=0)

predicted <- predict(model_rpart, test_data, type = "class")
table(predicted)

confusion.matrix <- prop.table(table(predicted, test_data$y))
confusion.matrix
confusionMatrix(predicted,test_data$y)

library(randomForest)

# train model
model_rf <- randomForest(y ~ ., data=train_data)
model_rf
```

```r
library(caret)
predicted_rf <- predict(model_rf, test_data)
table(predicted_rf)

confusion.matrix <- prop.table(table(predicted_rf, test_data$y))
confusion.matrix
confusionMatrix(predicted_rf,test_data$y)
```