Brunda Chouthoy
CSC 529: Case Study 1
UCI Bank Marketing Dataset – Who will subscribe for a term deposit?

Introduction

There are two main approaches for companies to promote their products/services: through mass campaigns, which target the public, and directed campaign, which targets only a specific group of people. Formal study shows that the efficiency of mass campaign is low. Usually less than 1% of the whole population will have positive response to the mass campaign. In contrast, Direct campaign focuses only on a small set of people who are believed to be interested in the product/service being marketed and thus would be much more efficient. In this case study, we focus only on the direct telemarketing data of a Portuguese banking institution.

For this case study, the main objective is to create a machine learning model using an ensemble classification technique like Random Forest and analyze how it performs compared to individual classification algorithms. My goal is to develop classification models that predict whether the customer will subscribe a term deposit based on 19 predictor variables and do a comparative assessment of performance for the individual data mining models and an ensemble classification approach such as Random Forest. After preprocessing the data, four data mining models were applied to compare the performance: k-nearest neighbor, Logistic Regression, Decision trees using RPart and an ensemble classification model Random Forest. Precision and Recall along with test accuracy will be used to compare various models.

Data Description and Understanding

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, to access if the product (bank term deposit) would be (or not) subscribed.
The binary classification goal is to build models to predict if the client will subscribe a term deposit(y).
Data Source: UCI Machine Learning Repository
Data url: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
Number of Instances: 41188 for bank-additional-full.csv
Number of Attributes: 20 + output attribute(y).
For each customer, the data records the result of telemarketing, either success or failure, as well as 20 telemarketing attributes. These attributes include four types of information:

- Customer features including age, job, marital status, education, default, housing and loan
- Phone call features including contact, month, day of the week and phone call duration
- 3 Social and economic factors including employment variation rate, consumer price index, consumer confidence index, 3 month Euribor rate and number of employees
- Other attributes including campaign, pdays, previous and poutcome.

Attribute Information:

Input variables:
#Bank client data
1 - age (numeric)
2 - job : type of job (categorical: 'admin.','bluecollar', 'entrepreneur', 'housemaid', 'management', 'retired','selfemployed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical:'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')
5 - default: has credit in default? (categorical: 'no', 'yes', 'unknown')
6 - housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
7 - loan: has personal loan? (categorical: 'no', 'yes', 'unknown')
related with the last contact of the current campaign:

8 - contact: contact communication type (categorical: 'cellular', 'telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon', 'tue', 'wed', 'thu', 'fri')
11 - duration: last contact duration, in seconds (numeric).
#other attributes:
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure', 'non-existent', 'success')

social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes', 'no')

Data Cleaning and Analysis:

After initial exploratory analysis, it could be determined that the bank telemarketing data is well organized and has NO missing data (shown in Table 2). Several categorical predictors take the value of 'unknown' but it should not be regarded as missing data; instead, the value 'unknown' should be treated as a level. Data distribution and summary is shown in Table 1.

```
> summary(bank_full)
      age                job            marital                    education        default         housing
 Min.   :17.00   admin.     :10422   divorced: 4612   university.degree  :12168   no     :32588   no     :18622
 1st Qu.:32.00   blue-collar: 9254   married :24928   high.school        : 9515   unknown: 8597   unknown:  990
 Median :38.00   technician : 6743   single  :11568   basic.9y           : 6045   yes    :    3   yes    :21576
 Mean   :40.02   services   : 3969   unknown :   80   professional.course: 5243
 3rd Qu.:47.00   management : 2924                    basic.4y           : 4176
 Max.   :98.00   retired    : 1720                    basic.6y           : 2292
                 (Other)    : 6156                    (Other)            : 1749
     loan           contact          month       day_of_week    duration         campaign          pdays
 no     :33950   cellular :26144   may    :13769   fri:7827    Min.   :   0.0   Min.   : 1.000   Min.   :  0.0
 unknown:  990   telephone:15044   jul    : 7174   mon:8514    1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0
 yes    : 6248                     aug    : 6178   thu:8623    Median : 180.0   Median : 2.000   Median :999.0
                                   jun    : 5318   tue:8090    Mean   : 258.3   Mean   : 2.568   Mean   :962.5
                                   nov    : 4101   wed:8134    3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0
                                   apr    : 2632               Max.   :4918.0   Max.   :56.000   Max.   :999.0
                                   (Other): 2016
   previous            poutcome      emp.var.rate      cons.price.idx  cons.conf.idx     euribor3m
 Min.   :0.000   failure    : 4252   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8   Min.   :0.634
 1st Qu.:0.000   nonexistent:35563   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.344
 Median :0.000   success    : 1373   Median : 1.10000   Median :93.75   Median :-41.8   Median :4.857
 Mean   :0.173                       Mean   : 0.08189   Mean   :93.58   Mean   :-40.5   Mean   :3.621
 3rd Qu.:0.000                       3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961
 Max.   :7.000                       Max.   : 1.40000   Max.   :94.77   Max.   :-26.9   Max.   :5.045

  nr.employed     y
 Min.   :4964   no :36548
 1st Qu.:5099   yes: 4640
 Median :5191
 Mean   :5167
 3rd Qu.:5228
 Max.   :5228
```

Table 1: Summary of all the attributes

```
> sapply(bank_full, function(bank_full) sum(is.na(bank_full)))
          age           job       marital     education       default       housing          loan
            0             0             0             0             0             0             0
      contact         month   day_of_week      duration      campaign         pdays      previous
            0             0             0             0             0             0             0
     poutcome emp.var.rate cons.price.idx cons.conf.idx     euribor3m   nr.employed             y
            0             0             0             0             0             0             0
```

Table 2: Missing values

Box plots were used for continuous variables to check for outliers and bar plots were used to check the distribution for categorical variables. Though some outliers are observed for the variables duration, Previous, and campaign (shown in figure 1, figure 2 and figure 3 respectively) they have not been removed keeping their significance into consideration. The outlier tends not to be a problem either, because most predictors are categorical and the response is binary. For the concept of outliers to be meaningful, distance must be defined first for the variable values, but the distance between the categorical values may not be possible to be defined. For example, for the categorical variable 'job', it is hard to tell if 'self-employed' is closer to 'housemaid' or a 'technician'.

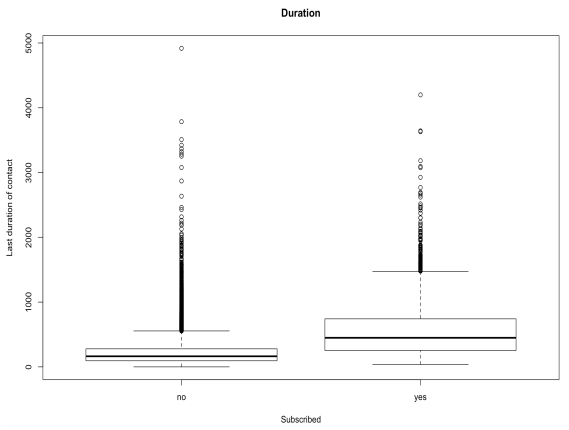


Figure 1: Box plot for duration

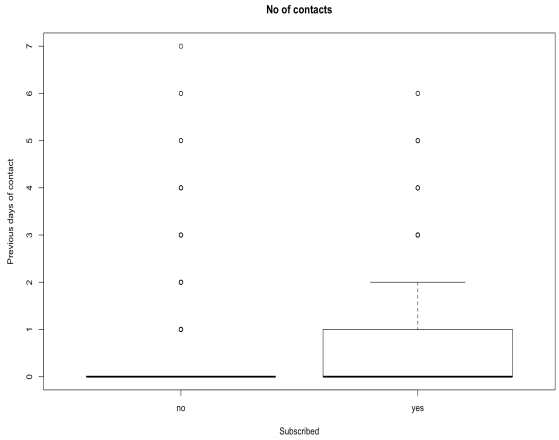


Figure 2: Box plot for previous contacts

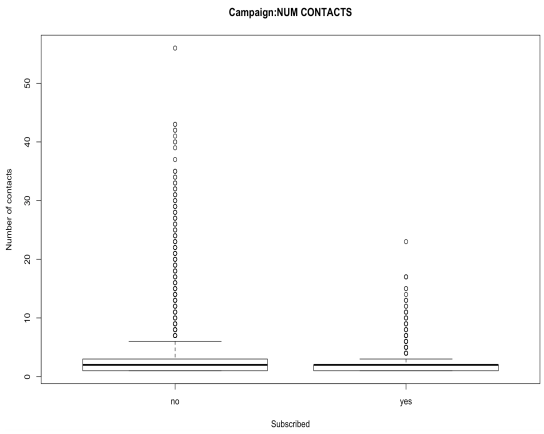


Figure 3: Box plot for Campaign

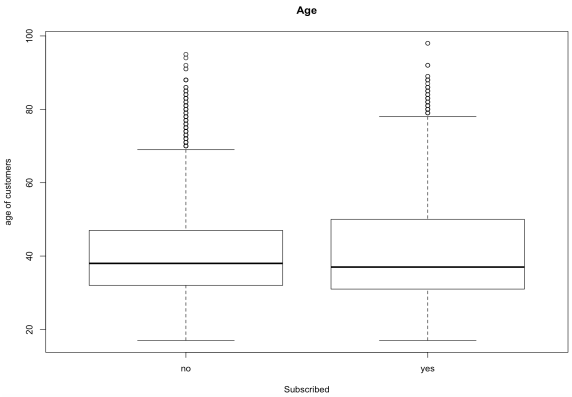
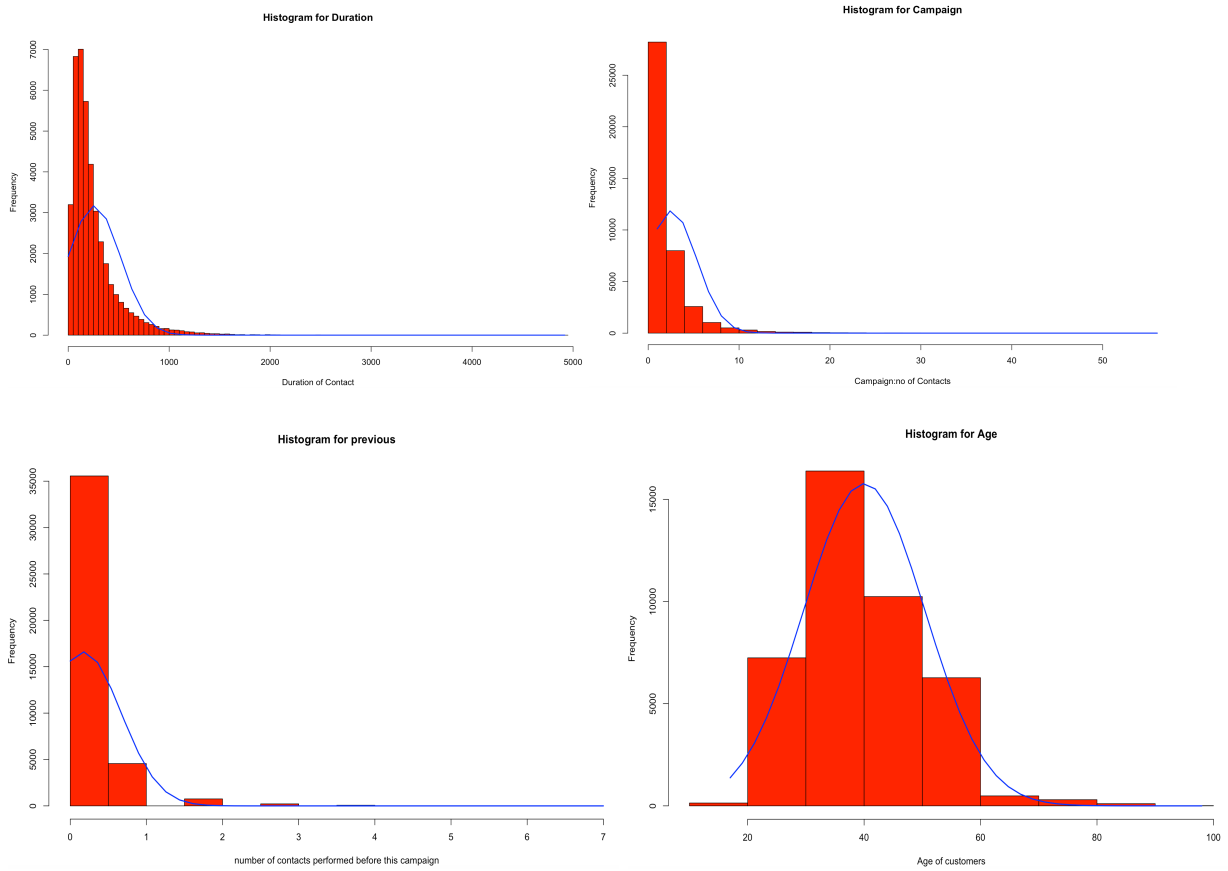


Figure 4: Box plot for Age

Histograms were used for continuous variables to accrue details about skewness, distribution etc. Duration is more skewed towards 0 to 1000 seconds, Campaign variable i.e. number of contacts performed during the

campaign is skewed towards 1. For the variable 'previous' - number of contacts performed before the campaign is skewed towards 0 i.e. many of the customers were not contacted previously.



Correlation Matrix (shown in Table-3) among input (or independent) continuous variables was calculated. It could be observed that no two variables are highly correlated.

```
> cor(bank_full.num)
                      bank_full.age bank_full.campaign bank_full.previous bank_full.pdays bank_full.emp.var.rate
bank_full.age          1.0000000000        0.00459358        0.02436474     -0.03436895         -0.0003706855
bank_full.campaign     0.0045935805        1.00000000       -0.07914147      0.05258357          0.1507538056
bank_full.previous     0.0243647409       -0.07914147        1.00000000     -0.58751386         -0.4204891094
bank_full.pdays       -0.0343689512        0.05258357       -0.58751386      1.00000000          0.2710041743
bank_full.emp.var.rate -0.0003706855        0.15075381       -0.42048911      0.27100417          1.0000000000
bank_full.cons.conf.idx  0.1293716142      -0.01373310       -0.05093635     -0.09134235          0.1960412681
bank_full.cons.price.idx 0.0008567150       0.12783591       -0.20312997      0.07888911          0.7753341708
                      bank_full.cons.conf.idx bank_full.cons.price.idx
bank_full.age                      0.12937161             0.000856715
bank_full.campaign                -0.01373310             0.127835912
bank_full.previous                -0.05093635            -0.203129967
bank_full.pdays                   -0.09134235             0.078889109
bank_full.emp.var.rate             0.19604127             0.775334171
bank_full.cons.conf.idx            1.00000000             0.058986182
bank_full.cons.price.idx           0.05898618             1.000000000
>
```

Table 3: Correlation matrix

Analysis of the result or target variable: Out of 41188 observations, 4640 or 11.2% of the customers subscribed for a term deposit and 36548 or 88.73% of the customers did not subscribe for a term deposit.

```
> #Forcing Yes to be the first factor
> bank_full$y <- factor( bank_full$y, levels=c("yes","no") )
> table(bank_full$y)

  yes    no
 4640 36548
> table(bank_full$y)/nrow(bank_full)

      yes        no
0.1126542 0.8873458
```

Attribute Selection: Since credit default is highly skewed towards No, the attribute 'default' was removed for further analysis. This makes the total number of predictors for analysis to be 19.

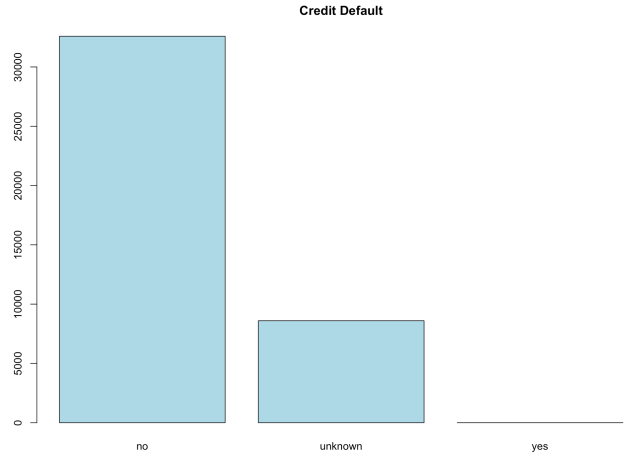


Figure 5: Bar Plot for Credit default

Data Preparation for k-nearest neighbors:

Prior to applying the k-NN model, the data attributes should be transformed since k-nearest neighbors only accept numerical attributes. We cannot use as.numeric() directly to convert factor variables to numeric as it has limitations. Thus, we convert the factors having character levels to numeric levels and then further convert these numeric level factors to numeric variables. Binary attributes have been mapped to {no = 1, yes= 2} values. Categorical attributes have been converted to discrete values. For example, 'education' level has been mapped as {primary = 1, secondary = 2, tertiary = 3} and 'poutcome' is mapped as {failure = 1, nonexistent = 2, success = 3}. Furthermore, the numeric variables were normalized - this feature is of paramount importance since the scale used for the values for each variable might be different. The best practice is to normalize the data and transform all the values to a common scale. Table 4 shows a snapshot of the classes of variable after transformation and normalization of the numeric data.

```
> str(bank_full_normalized)
'data.frame':   41188 obs. of  20 variables:
 $ age           : num  0.481 0.494 0.247 0.284 0.481 ...
 $ job           : num  0.273 0.636 0.636 0 0.636 ...
 $ marital       : num  0.333 0.333 0.333 0.333 0.333 ...
 $ education     : num  0 0.429 0.429 0.143 0.429 ...
 $ housing       : num  0 0 0.5 0 0 0 0 0.5 0.5 ...
 $ loan          : num  0 0 0 0 0.5 0 0 0 0 0 ...
 $ contact       : num  1 1 1 1 1 1 1 1 1 1 ...
 $ month         : num  0.727 0.727 0.727 0.727 0.727 ...
 $ day_of_week   : num  0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 ...
 $ duration      : num  0.0531 0.0303 0.046 0.0307 0.0624 ...
 $ campaign      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ pdays         : num  1 1 1 1 1 1 1 1 1 1 ...
 $ previous      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome      : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
 $ emp.var.rate  : num  0.938 0.938 0.938 0.938 0.938 ...
 $ cons.price.idx: num  0.699 0.699 0.699 0.699 0.699 ...
 $ cons.conf.idx : num  0.603 0.603 0.603 0.603 0.603 ...
 $ euribor3m     : num  0.957 0.957 0.957 0.957 0.957 ...
 $ nr.employed   : num  0.86 0.86 0.86 0.86 0.86 ...
 $ y             : num  1 1 1 1 1 1 1 1 1 1 ...
```

Table 4: Data after transformation and normalization

Brunda Chouthoy
CSC 529: Case Study 1
UCI Bank Marketing Dataset – Who will subscribe for a term deposit?

Experimental Results:

The following machine learning algorithms were used for this case study:

1. Decision Trees using the Rpart package
2. K- nearest neighbors
3. Logistic Regression
4. Random Forest

Decision Tree using rpart package:

- Training and testing sets were created using the 80:20 rule, which can be used to train the dataset and test those values with the test set.
  No of records in training -> 32950
  No of records in testing -> 8238
- A Decision Tree is a robust and transparent Machine Learning model. The tree starts with a single node and then branches out, with a decision being made at every branch point. The model can be used to predict whether a variable would have mattered in the customer's decision to subscribe or not to the bank's term deposit.
- I have used the rpart package to build the decision tree in R and the fancyRpartPlot function in the 'rattle' package to plot the tree.
- The decision tree model for the bank marketing data is shown in Figure 6.

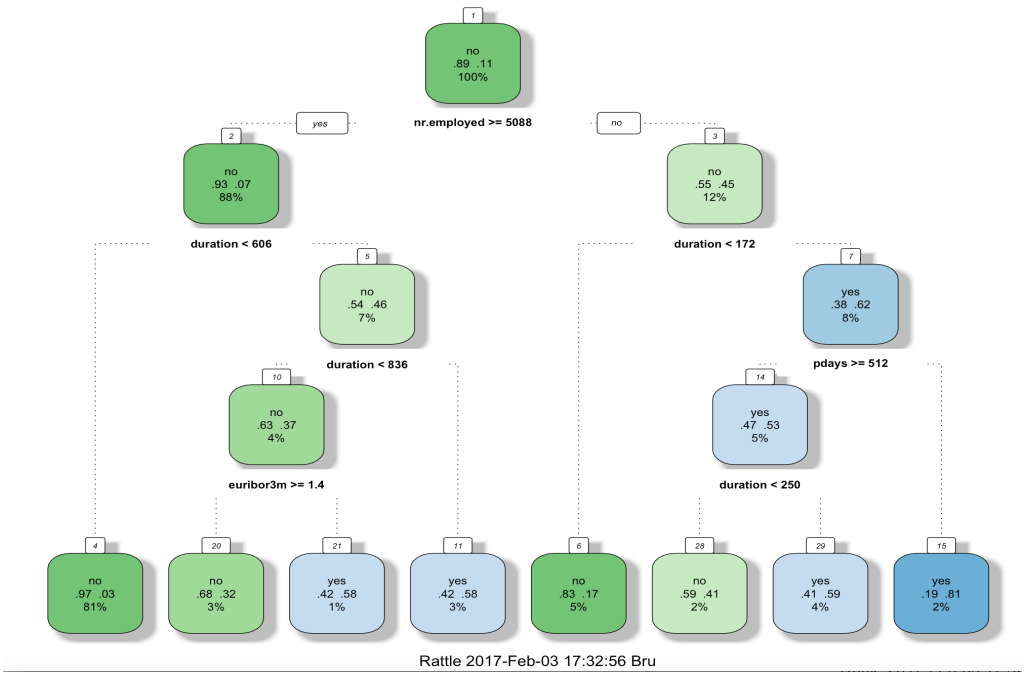


Figure 6: Decision tree for Bank Marketing data

Interpretation of the decision tree:

a) At the top node of the decision tree we see that 89% of the people are employed, whereas 11% are unemployed and it can be interpreted that the top node represents 100% of the customer base (Node 1).

b) Suppose we want to look at the employed passengers, the next step is to see how many of them on the call for more than 606 seconds and the tree is divided accordingly. We can interpret that 93% of the people were on the call for less than 606 seconds whereas 7% of them were on the call for more time. (Node 2)
c) Next we can see look at the people who were on the call for lesser duration. We can see that 97% of them did not subscribe whereas 3% of them did. (Node 4)
d) Next step, we look at the people who were on the call for more duration. We see that 54% of them were on the call for more than 836 seconds and 46% of them were there for less duration. (Node 5)
e) Similarly, among the people who were on the call for less than 836 seconds, 68% did not subscribe, whereas 32% subscribed (Node 20). Whereas, among the 42% who were on the call for more than 836 seconds 43% subscribed and 58% did not subscribe (Node 21).

The variable importance table of the decision tree model on the training data is shown below:

```
Variable importance
      duration    nr.employed      euribor3m   emp.var.rate  cons.conf.idx cons.price.idx          month
            24             19             17             11             10              9              5
         pdays       poutcome       previous
             1              1              1
```

Table 5: Variable importance of the decision tree model

The accuracy of the model on the train data is **91.34%.**
The model was validated on the test data and the accuracy was calculated. The accuracy of the model on the test data was **91.61%.**

```
> t <- table(pred_rpart,train_tree$y)
> confusionMatrix(t)
Confusion Matrix and Statistics


pred_rpart    no   yes
       no  28026  1667
       yes  1186  2071

               Accuracy : 0.9134
                 95% CI : (0.9103, 0.9164)
    No Information Rate : 0.8866
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.544
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9594
            Specificity : 0.5540
         Pos Pred Value : 0.9439
         Neg Pred Value : 0.6359
             Prevalence : 0.8866
         Detection Rate : 0.8506
   Detection Prevalence : 0.9012
      Balanced Accuracy : 0.7567

       'Positive' Class : no
```

Table 7: Confusion Matrix for the train data

```
> t <- table(pred_rpart,test_tree$y)
> confusionMatrix(t)
Confusion Matrix and Statistics


pred_rpart   no  yes
       no  7039  394
       yes  297  508

               Accuracy : 0.9161
                 95% CI : (0.9099, 0.922)
    No Information Rate : 0.8905
    P-Value [Acc > NIR] : 6.518e-15

                  Kappa : 0.5486
 Mcnemar's Test P-Value : 0.0002602

            Sensitivity : 0.9595
            Specificity : 0.5632
         Pos Pred Value : 0.9470
         Neg Pred Value : 0.6311
             Prevalence : 0.8905
         Detection Rate : 0.8545
   Detection Prevalence : 0.9023
      Balanced Accuracy : 0.7614

       'Positive' Class : no
```

Table 6: Confusion Matrix for the test data

K- nearest neighbors

- k nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. This algorithm segregates unlabeled data points into well-defined groups.
- After the data preparation step for k-NN, Training and testing sets were created using the 80:20 rule, which can be used to train the dataset and test those values with the test set.
  No of records in training -> 32950
  No of records in testing -> 8238
- The knn () function needs to be used to train a model for which I used the package 'class'. The knn() function identifies the k-nearest neighbors using Euclidean distance where k is a user-specified number.
- To check the accuracy of the predicted values as to whether they match up with the known values of y, I used the CrossTable() function available in the package 'gmodels'.
- The confusion matrix and the results for k-NN when k=25 is shown in table 7

```
> knn_bank_25 <- knn(train = train_knn, test = test_knn,cl = train_label, k=25)
> CrossTable(x=test_label, y=knn_bank_25, prop.chisq = FALSE)


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


Total Observations in Table:  8238


             | knn_bank_25
  test_label |         1 |         2 | Row Total |
-------------|-----------|-----------|-----------|
           1 |      7192 |       124 |      7316 |
             |     0.983 |     0.017 |     0.888 |
             |     0.909 |     0.378 |           |
             |     0.873 |     0.015 |           |
-------------|-----------|-----------|-----------|
           2 |       718 |       204 |       922 |
             |     0.779 |     0.221 |     0.112 |
             |     0.091 |     0.622 |           |
             |     0.087 |     0.025 |           |
-------------|-----------|-----------|-----------|
Column Total |      7910 |       328 |      8238 |
             |     0.960 |     0.040 |           |
-------------|-----------|-----------|-----------|
```

Table 8: Confusion matrix for k-NN when k=25

The test data consisted of 8238 observations. Out of which 7192 cases have been accurately predicted (TN->True Negatives) as 'no' in nature. Also, 204 observations were accurately predicted (TP-> True Positives) as 'Yes' in nature. There were 718 cases of False Negatives (FN) meaning 718 cases were recorded which are Yes in nature but got predicted as No. The FN's if any poses a potential threat for the same reason and the focus to increase the accuracy of the model is to reduce FN's. There were 124 cases of False Positives (FP) meaning 124 cases were 'No' in nature but got predicted as 'Yes'.

The total accuracy of the model when k=25 is (7192+204/8238) i.e. **89.779%**

- The confusion matrix and the results for k-NN when k=50 is shown in table 8

```
> knn_bank_50 <- knn(train = train_knn, test = test_knn,cl = train_label, k=50)
> CrossTable(x=test_label, y=knn_bank_50, prop.chisq = FALSE)


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


Total Observations in Table:  8238


             | knn_bank_50
  test_label |         1 |         2 | Row Total |
-------------|-----------|-----------|-----------|
           1 |      7209 |       107 |      7316 |
             |     0.985 |     0.015 |     0.888 |
             |     0.908 |     0.360 |           |
             |     0.875 |     0.013 |           |
-------------|-----------|-----------|-----------|
           2 |       732 |       190 |       922 |
             |     0.794 |     0.206 |     0.112 |
             |     0.092 |     0.640 |           |
             |     0.089 |     0.023 |           |
-------------|-----------|-----------|-----------|
Column Total |      7941 |       297 |      8238 |
             |     0.964 |     0.036 |           |
-------------|-----------|-----------|-----------|
```

Table 9: Confusion matrix for k-NN when k=50

The total accuracy of the model when k=25 is (7209+190/8238) i.e. **89.815%**

Logistic Regression

- Training and testing sets were created using the 80:20 rule, which can be used to train the dataset and test those values with the test set.
  No of records in training -> 32950
  No of records in testing -> 8238
- Logistic regression is a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. It predicts the probability of occurrence of an event by fitting data to a logit function.
- The confusion matrix and ROC plot with results is shown in Table 9 and Figure respectively.

```
> library(pROC)
> ROC=roc(y~test.prob,data=test_log)
> plot(ROC)  ## Here cut off value = 0.5 has been selected.

Call:
roc.formula(formula = y ~ test.prob, data = test_log)

Data: test.prob in 7316 controls (y no) < 922 cases (y yes).
Area under the curve: 0.9345
> test.prediction<-cut(test.prob,c(-Inf,0.5,Inf),labels=c("no","yes"))
> summary(test.prediction)
  no  yes
7655  583
> ##Confusion Matrix
> table(test.prediction,test_log$y)

test.prediction   no  yes
            no  7134  521
            yes  182  401
```

Table 10:Confusion Matrix for the Log regression model

Based on the confusion matrix, the accuracy of the model is (7134+401/8238) i.e. 91.46%.

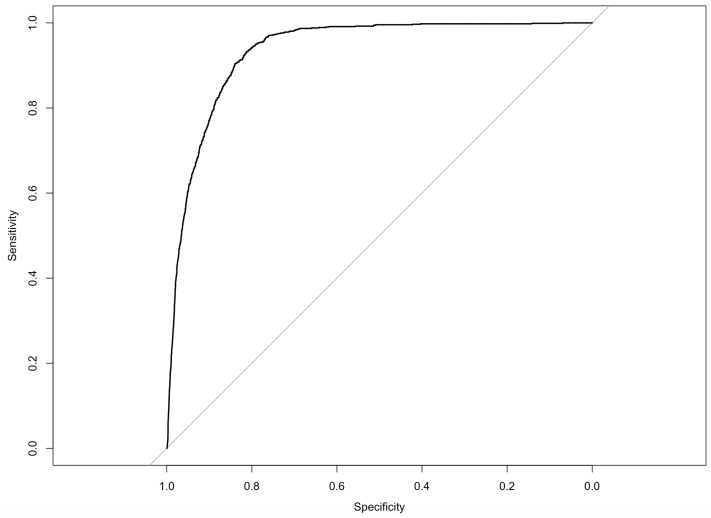


Figure 7: ROC curve for the logistic regression model

Receiver Operating Characteristic(ROC) summarizes the model's performance by evaluating the tradeoffs between true positive rate (sensitivity) and false positive rate (1- specificity). ROC curve for the logistic regression model is shown in Figure 7.
For plotting ROC, I have assumed p > 0.5 since we are more concerned about success rate. ROC summarizes the predictive power for all possible values of p > 0.5. The area under curve (AUC), referred to as index of accuracy(A), is a perfect performance metric for ROC curve. Higher the area under curve, better the prediction power of the model.

Random Forest

- Random Forest is a versatile machine learning method capable of performing both regression and classification tasks. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.
- In Random Forest, we grow multiple trees and to classify a new object based on attributes, each tree gives a classification and each tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).
- Random Forest involves sampling of the input data with replacement called as bootstrap sampling. Here one third of the data is not used for training and can be used to testing. These are called the out of bag samples. Error estimated on these out of bag samples is known as out of bag error. Study of error estimates by Out of bag, gives evidence to show that the out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set aside test set.
- No of records in training -> 33007 -- Sampling with Replacement
  No of records in testing -> 8181

```
> table(bank_train$y)

   no   yes
29287  3720
> table(bank_train$y)/nrow(bank_train)

       no       yes
0.8872966 0.1127034
>
> table(bank_test$y)

  no  yes
7261  920
> table(bank_test$y)/nrow(bank_test)

       no       yes
```

Table 11: Train and test data distribution

- Both the train and test data sets have similar target variable distribution. This is just a sample validation.

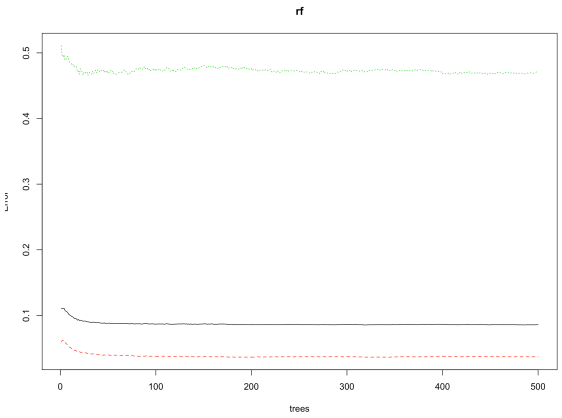
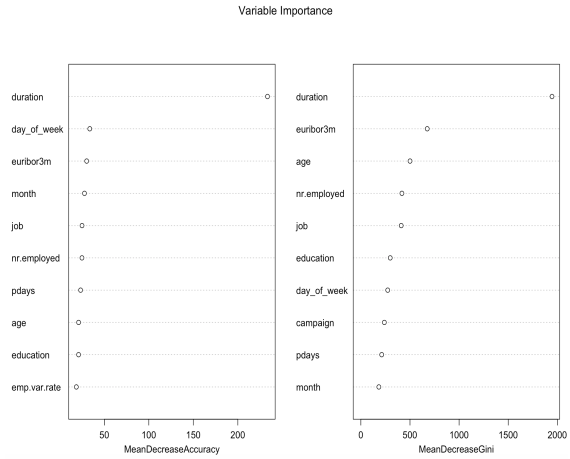


Figure 8: Trees Vs Error

- 500 decision trees or a forest has been built using the Random Forest algorithm based learning and displayed in Figure 8. I have plotted the error rate across decision trees. The plot seems to indicate that after 100 decision trees, there is not a significant reduction in error rate.
- Variable importance plot is also a useful tool and can be plotted using varImpPlot function. Top 10 variables are selected and plotted based on Model Accuracy and Gini value. The Table 11 shows a table with decreasing order of importance based on a measure (1 for model accuracy and 2 node impurity)




```
> var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),]
              MeanDecreaseGini      Variables
duration           1945.71377       duration
euribor3m           675.63573      euribor3m
age                 500.42876            age
nr.employed         417.89767    nr.employed
job                 410.64137            job
education           298.97798      education
day_of_week         273.33698    day_of_week
campaign            239.07217       campaign
pdays               212.13216          pdays
month               182.95978          month
poutcome            168.90765       poutcome
cons.conf.idx       156.00581  cons.conf.idx
cons.price.idx      134.51768 cons.price.idx
emp.var.rate        132.28852   emp.var.rate
marital             127.54080        marital
housing             103.58671        housing
loan                 80.79012           loan
previous             80.28251       previous
contact              58.18331        contact
>
```

Figure 9: Variable importance plot

- Since we want to know which variable includes the largest amount of information, we should focus on the mean decrease of Gini Index, which is a measurement of statistical dispersion. It can be seen that the most important variable is duration, which is the last contact duration; the second is euribor3m which is euribor 3 month rate; the third is age; and fourth is nr.employed, which is the number of employees in the bank.
- confusionMatrix function from caret package can be used for creating confusion matrix based on actual response variable and predicted value.
- The accuracy of the training data 99.75%. Now we can predict response for the validation sample and calculate model accuracy for the sample. Table 11 shows the confusion matrix and accuracy for the training data.
- The accuracy of the testing data 91.61%, which is still significantly higher. Table 12 shows the confusion matrix and accuracy for the training data.

```
> # Predicting response variable
> bank_test$predicted.response <- predict(rf ,bank_test)
>
> # Create Confusion Matrix
> confusionMatrix(data=bank_test$predicted.response,
+                 reference=bank_test$y,
+                 positive='no')
Confusion Matrix and Statistics

          Reference
Predicti confusionMatrix(data, ...)
       no  7000  425
       yes  261  495

               Accuracy : 0.9161
                 95% CI : (0.9099, 0.9221)
    No Information Rate : 0.8875
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5445
 Mcnemar's Test P-Value : 4.866e-10

            Sensitivity : 0.9641
            Specificity : 0.5380
         Pos Pred Value : 0.9428
         Neg Pred Value : 0.6548
             Prevalence : 0.8875
         Detection Rate : 0.8556
   Detection Prevalence : 0.9076
      Balanced Accuracy : 0.7510

       'Positive' Class : no
```

Table 13:Confusion Matrix for Test data

```
> # Create Confusion Matrix
> confusionMatrix(data=bank_train$predicted.response,
+                 reference=bank_train$y,
+                 positive='no')
Confusion Matrix and Statistics

          Reference
Prediction    no   yes
       no  29286    82
       yes     1  3638

               Accuracy : 0.9975
                 95% CI : (0.9969, 0.998)
    No Information Rate : 0.8873
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9873
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.0000
            Specificity : 0.9780
         Pos Pred Value : 0.9972
         Neg Pred Value : 0.9997
             Prevalence : 0.8873
         Detection Rate : 0.8873
   Detection Prevalence : 0.8898
      Balanced Accuracy : 0.9890

       'Positive' Class : no
```

Table 12: Confusion Matrix for train data

Experimental Analysis:

Table 14 shows the results consolidated from all the four models.

Testing Data	Accuracy	Precision	Recall or Sensitivity
1. Decision Trees using the Rpart package	0.9161 or 91.61%	0.5632	0.9595
2. K- nearest neighbors (k=50)	0.8981 or 89.81%	0.6397	0.9078
3. Logistic Regression	0.9146 or 91.46%	0.4349	0.9751
4. Random Forest (n=500 trees)	0.9161 or 91.61%	0.538	0.9641

- As per the results above, the accuracy of the models range from 89%-91.61% for the testing dataset. Precision and Recall along with test accuracy will be used to compare various models. Based on the different parameters, Decision tree model and Random Forest ensemble have the best possible values. The recall or Sensitivity i.e. the proportion of actual positive cases which are correctly identified is highest (0.9641) for the ensemble classifier Random Forest. However, the Random Forest does not give significantly better results compared to the others.
- For all the methods except log regression, the precision is higher than 50%, which is good. Note that the percentage of positive outcomes is 11%. So, compared to mass (random) campaign, our learning procedure seems to be much more efficient for the bank.
- As for False Positive Rate(FPR), one of our objectives is to build a model, which can identify almost all (i.e. 99%) of the contacts who will eventually subscribe a term deposit, while keeping a high overall prediction accuracy. With these models, we may significantly reduce the workload and costs,

by researching over a much smaller group in which people are predicted to subscribe a term deposit instead of the massive population.

- According to the plots and variable importance tables from the methods, we can conclude that the most influential variables are duration, nr.employed, euribor3m, and emp.var.rate.
- The attribute "duration" has positive effect on people saying "yes". This is because the longer the conversations on the phone, the higher interest the customer will show to the term deposit. "nr.employed", which is the number of employees in the bank, has positive effect for turning people to subscribe the term deposit. This can be because the more employees the bank has, the more influential and prestigious this bank is. "euribor3m" is another important variable, which denotes the euribor 3-month rate. This indicator is based on the average interbank interest rates in Eurozone. It also has positive effect since the higher the interest rate the more willingly customer will spend their money on financial tools. Employment variation rate (emp.var.rate) has negative influence, which means the change of the employment rate will make customers less likely to subscribe a term deposit. This makes sense because the employment rate is an indicator of the macro economy. A stable employment rate denotes a stable economic environment in which people are more confident to make their investment.
- Decision tree dominates in two measurements (Accuracy and Precision) and ranked 2nd in Recall, so it's the most powerful model. Random forest has a similar result to the decision tree with slightly low precision. Logistic regression has an acceptable performance. Despite a low result for Precision parameter, it provides a practical way to make inferences. k-NN, as a baseline model, has the worst performance.

Conclusion:

- For this case study, the main objective is to create a machine learning model using an ensemble classification technique like Random Forest and analyze how it performs compared to individual classification algorithms.
- Individual classification algorithms like k-NN, logistic regression and Decision tree were applied to conduct a performance assessment with an ensemble classification technique like Random Forest.
- For each customer, the data records the result of telemarketing, either success or failure, as well as 20 telemarketing attributes. Since credit default is highly skewed towards No, the attribute 'default' was removed for further analysis. This makes the total number of predictors for analysis to be 19.
- It could be observed that out of 41188 observations, 4640 or 11.2% of the customers subscribed for a term deposit and 36548 or 88.73% of the customers did not subscribe for a term deposit.
- After initial analysis, it could be determined that the bank telemarketing data is well organized and has NO missing data. Many different variables were examined as a part of the exploratory analysis of data using Boxplots, histograms and bar charts to learn about outliers, distribution etc. Correlation Matrix among input (or independent) continuous variables was calculated and could be observed that no two variables are highly correlated.
- Data cleaning and preparation was done to structure the data in a way that can be used to apply the various data mining models.
- Data was modeled with four different algorithms and the performance metrics were documented along with interpretation, plots and tables.
- The data was sampled with Replacement for Random Forest called as bootstrap sampling. The 500 trees Vs error plot indicated that after 100 decision trees, there was not a significant reduction in error rate.
- Experimental results were then analyzed considering different performance metrics, the accuracy of the models range from 89%-91.61% for the testing dataset. Precision and Recall along with test accuracy was used to compare various models.

Brunda Chouthoy
CSC 529: Case Study 1
UCI Bank Marketing Dataset – Who will subscribe for a term deposit?

- According to the plots and variable importance tables from the methods, we can conclude that the most influential variables are duration, nr.employed, euribor3m, and emp.var.rate.
- Decision tree model and Random Forest ensemble have the best possible values for accuracy. The recall or Sensitivity i.e. the proportion of actual positive cases which are correctly identified is highest (0.9641) for the ensemble classifier Random Forest. However, the Random Forest does not give significantly better results compared to the others.
- Based on the above analysis, Decision tree dominates in two measurements (Accuracy and Precision) and ranked 2nd in Recall, so it's the most powerful model. Random forest has a similar result to the decision tree with slightly low precision. Logistic regression has an acceptable performance. Despite a low result for Precision parameter, it provides a practical way to make inferences. k-NN, as a baseline model, has the worst performance.
- Bank direct marketing and business decisions are more important than ever for preserving the relationship with the best customer. Data mining and predictive analytics are of immense help in such marketing strategies. Its applications are influential in almost every field containing complex data and large procedures. It has proven the ability to reduce the number of false positives and false negative decisions.