Bank Marketing analysis

**by**

Harnoor Rangi - Naveed Sultan -Abhishek Pathak- Pravin Kale - Qumeng Wu

Introduction

Our group has formed an key analysis on the Bank Marketing dataset to identify which customers would subscribe to a long term deposit account. Data analytical methods have been applied to identify customers and determine an effective telemarketing strategy for the Bank marketing dataset. An accurate conclusion has been developed by applying the three stages of the analytical methods which include data preparation, predictive modeling and post predictive modeling.

Bank Marketing Dataset

The Bank Marketing dataset can be very important depending upon the kind of variables that are included in the dataset. It can be used to predict and forecast the number of clients that have subscribed a term deposit or not.

**Tools: R – Studio and Weka**

**Data Preparation**

# **Explanation**

The data is related with direct marketing campaigns of a banking institution. The marketing campaigns were based on phone calls.

**The Goal**

Apply analytical methods to determine customer's subscription to a long term deposit account.

**Dataset Initial Impressions**

* Total 4521 records
* 7 numeric attributes: age, balance, day, duration, campaign, pdays, previous
* 10 Factors:
* 6 multi-valued categorical attributes: job, marital, education, contact, month, poutcome
* 3 yes/no binary attributes: default, housing, loan
* 1 class attribute y
* No missing values: Preprocessing should be easier, no data cleaning needed
* imbalance in the dataset with greater amount of No’s in class attribute

**The variables are as follows:**

* 1 - age (numeric)
* 2 - job: type of job (categorical: 'admin.', 'unknown', 'unemployed', 'management', 'housemaid', 'entrepreneur', 'student', 'blue-collar’, ‘self-employed','retired','technician','services')
* 3 - marital: marital status (categorical: 'married', 'divorced', 'single'; note: 'divorced' means divorced or widowed)
* 4 - education (categorical: 'unknown', 'secondary', 'primary', 'tertiary')
* 5 - default: has credit in default? (binary: 'yes', 'no')
* 6 - balance: average yearly balance, in euros (numeric)
* 7 - housing: has housing loan? (binary: 'yes', 'no')
* 8 - loan: has personal loan? (binary: 'yes', 'no') , related with the last contact of the current campaign:
* 9 - contact: contact communication type (categorical: 'unknown', 'telephone', 'cellular')
* 10 - day: last contact day of the month (numeric)
* 11 - month: last contact month of year (categorical: 'Jan', 'feb', 'mar', ..., 'nov', 'dec')
* 12 - duration: last contact duration, in seconds (numeric), other attributes:
* 13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
* 14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
* 15 - previous: number of contacts performed before this campaign and for this client (numeric)
* 16 - poutcome: outcome of the previous marketing campaign (categorical: 'unknown', 'other', 'failure', 'success')
* Output variable (desired target):
* 17 - y - has the client subscribed a term deposit? (binary: 'yes', 'no')

## Data Preparation

# Read the dataset

bmd<-read.csv("bank.csv",header = TRUE)

## Attribute type:

**Age Job Marital Education Default**

**Discrete-Quantitative Categorical Categorical Categorical Categorical**

**Balance Housing Loan contact**

**Discrete-Quantitative Categorical Categorical Categorical**

**Day Month Duration**

**Ordinal Categorical Descrete-Quantitative**

**Campaign Pday Previous Poutcome Y**

**Ordinal Continuous- Ordinal Categorical Categorical**

**quantitative**

## Summary of all Attributes:

summary(bmd)

age job marital education default

Min. :19.00 management :969 divorced: 528 primary: 678 no :4445

1st Qu.:33.00 blue-collar:946 married :2797 secondary:2306 yes: 76

Median :39.00 technician :768 single :1196 tertiary :1350

Mean :41.17 admin. :478 unknown: 187

3rd Qu.:49.00 services :417

Max. :87.00 retired :230

SD :3.255716 (Other) :713

balance housing loan contact day

Min. : -3313 no :1962 no :3830 cellular :2896 Min. : 1.00

1st Qu.: 69 yes:2559 yes: 691 telephone: 301 1st Qu.: 9.00

Median : 444 unknown :1324 Median :16.00

Mean : 1423 Mean :15.92

3rd Qu.: 1480 3rd Qu.:21.00

Max. :71188 Max. :31.00

SD :3009.638 SD :8.247667

month duration campaign pdays previous

may :1398 Min. : 4 Min. : 1.000 Min. : -1.00 Min. : 0.0000

jul : 706 1st Qu.: 104 1st Qu.: 1.000 1st Qu.: -1.00 1st Qu.: 0.0000

aug : 633 Median : 185 Median : 2.000 Median : -1.00 Median : 0.0000

jun : 531 Mean : 264 Mean : 2.794 Mean : 39.77 Mean : 0.5426

nov : 389 3rd Qu.: 329 3rd Qu.: 3.000 3rd Qu.: -1.00 3rd Qu.: 0.0000

apr : 293 Max. :3025 Max. :50.000 Max. :871.00 Max. :25.0000

(Other): 571 SD :259.8566 SD :3.109807 SD :100.1211

poutcome y

failure: 490 no :4000

other : 197 yes: 521

success: 129

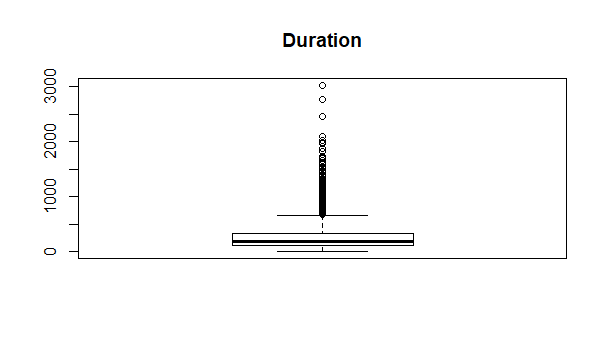
unknown:3705

**Determine any outlier values (boxplots)**

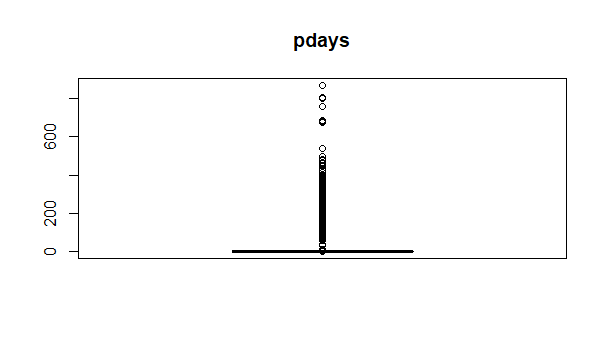
## bmdage<-boxplot(bmd$age, main = "Age",col = "red")

## 

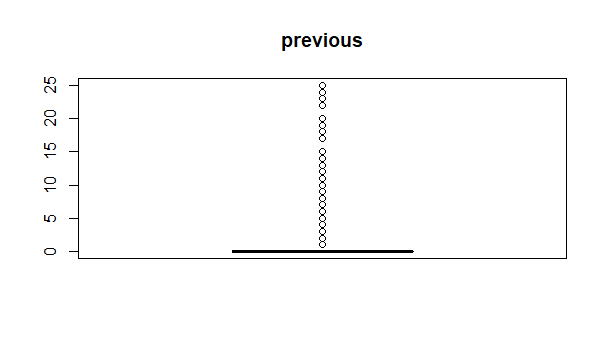
**bmdduration<-boxplot(bmd$duration, main = "Duration")**



**bmdpdays<-boxplot(bmd$pdays, main = "pdays")**



**bmdprevious<-boxplot (bmd$previous, main = "previous")**



boxplot(bmd$age~bmd$job,rm.na=TRUE,

data=bmd,

main="Boxplot for job status as per the age",

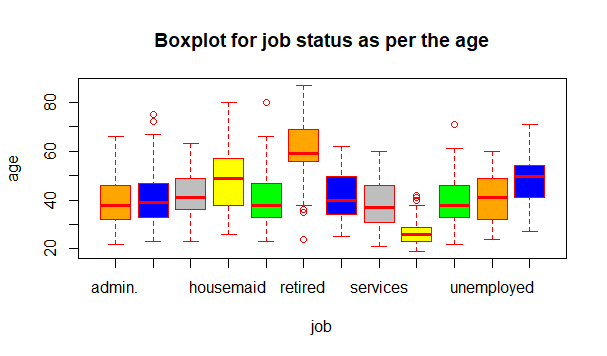
xlab="job",

ylab="age",

col=c("orange","blue","grey","yellow","green"),

border="red"

)



#marital status as per the age

boxplot(bmd$age~bmd$marital,rm.na=TRUE,

data=bmd,

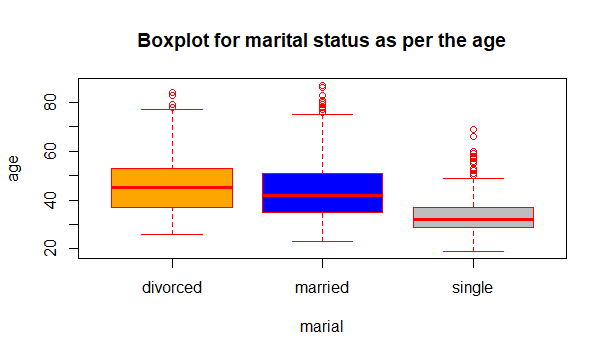
main="Boxplot for marital status as per the age",

xlab="marial",

ylab="age",

col=c("orange","blue","grey","yellow","green"),

border="red"



#education status as per the age

boxplot(bmd$age~bmd$education,rm.na=TRUE,

data=bmd,

main="Boxplot for education status as per the age",

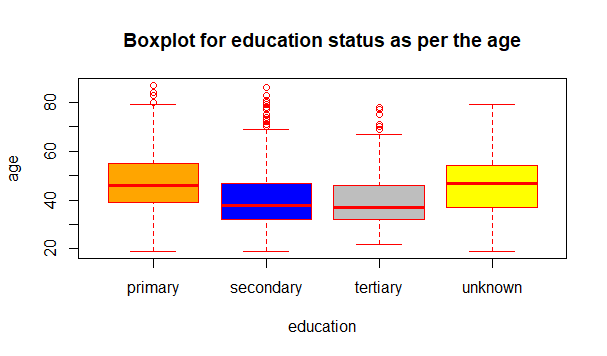
xlab="education",

ylab="age",

col=c("orange","blue","grey","yellow","green"),

border="red"

)



#default status as per the age

boxplot(bmd$age~bmd$default,rm.na=TRUE,

data=bmd,

main="Boxplot for default status as per the age",

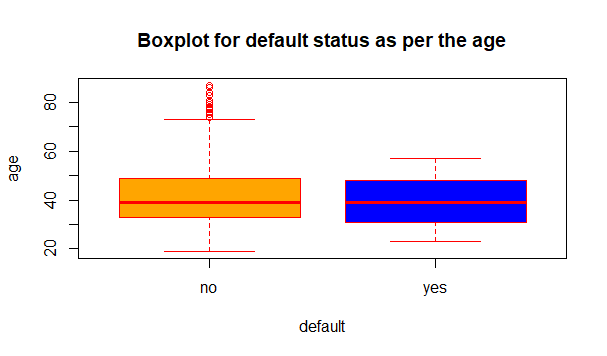
xlab="default",

ylab="age",

col=c("orange","blue","grey","yellow","green","grey"),

border="red"

)



#age status as per the balance

boxplot(bmd$balance~bmd$age,rm.na=TRUE,

data=bmd,

main="Boxplot for age status as per the balance",

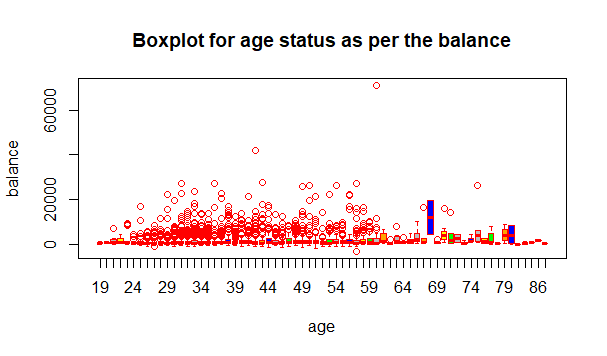
xlab="age",

ylab="balance",

col=c("orange","blue","grey","yellow","green","grey"),

border="red"

)

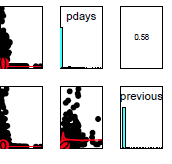


**Box Plot Summary:**

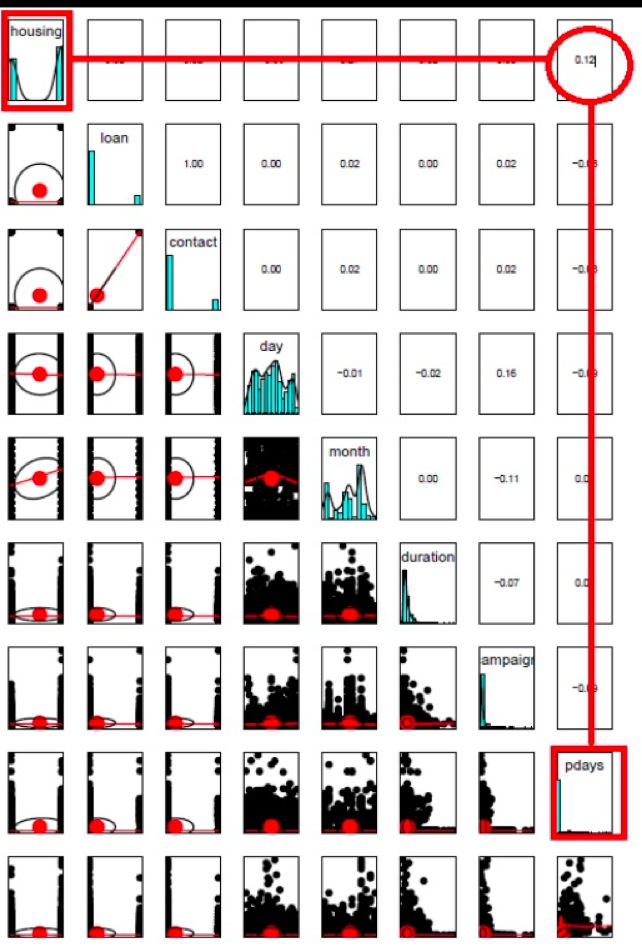
Box plot is great tool for displaying large data in a relatively small area to analyze the data at a first glance. Boxplot can give the user many insights to the user such as the quartiles, the the median, the minimum and max values. In the bank data, it is vital in determining the outliers. As can be seen many of the bank data attributes outliers. Due to the large bank data, the outliers did not affect the results nor the assumptions made, as a result the group choose to keep the outliers part of the data.

Attribute Correlation

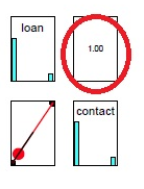
Correlation 1:



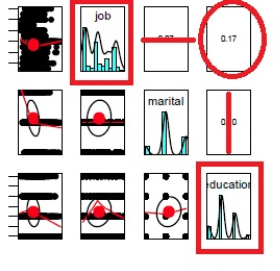
Correlation 2:



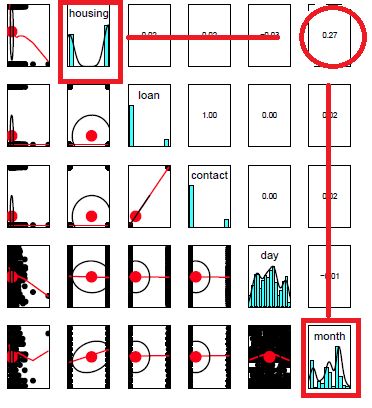
Correlation 3:



Correlation 4:



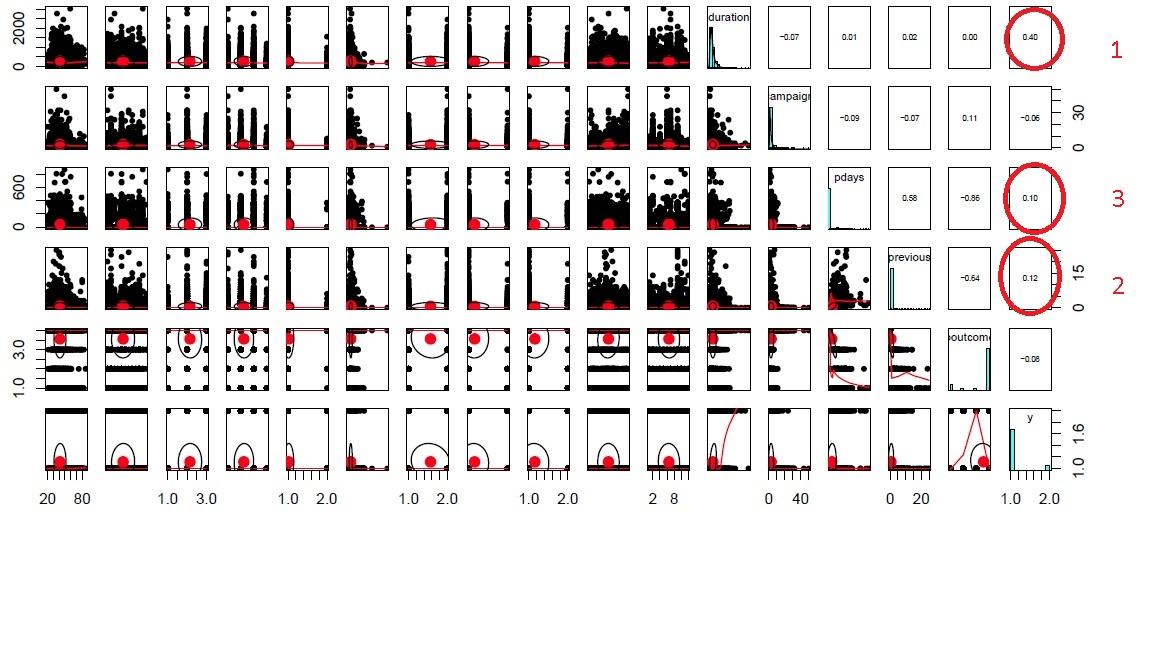
Correlation 5:



**Summary Attribute correlation:**

The above diagrams show case the attribute relationships excluding the class attribute. The importance of above diagrams show case the positive union between the attributes, indicating their union affect. The closer the values to 1, the greater the positive correlation between the attributes. It essentially is measuring the linear association between the two numerical attributes. For example, correlation 3, shows a perfect correlation score of positive 1. It means that they have a positive linear relationship, indicating the effect they have one each other as one increases in a linear fashion.

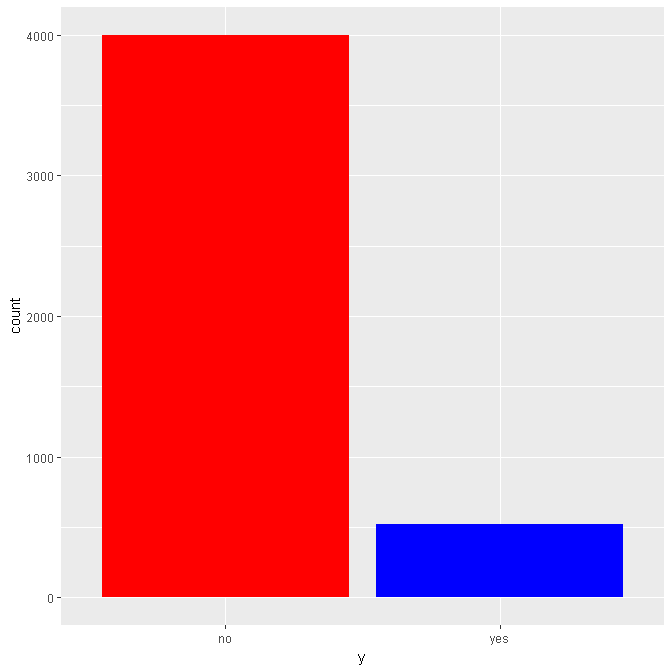
Class Correlation:



**Class correlation Summary:**

The above figure showcases the class attribute correlation with other attributes. The same rule applies, as their is a positive correlation number and is closer to the value 1, it indicates a positive linear association between the attribute and the class attribute. The duration attribute showcase the greatest amount of positive linear relation which amounts to a high effect of the duration attribute upon the class attribute y. It is very important to know which attributes have an high positive linear relationship to the class attribute as it allows the analyst to remove attributes that may make the results of the data unclear.

**Imbalance Class**

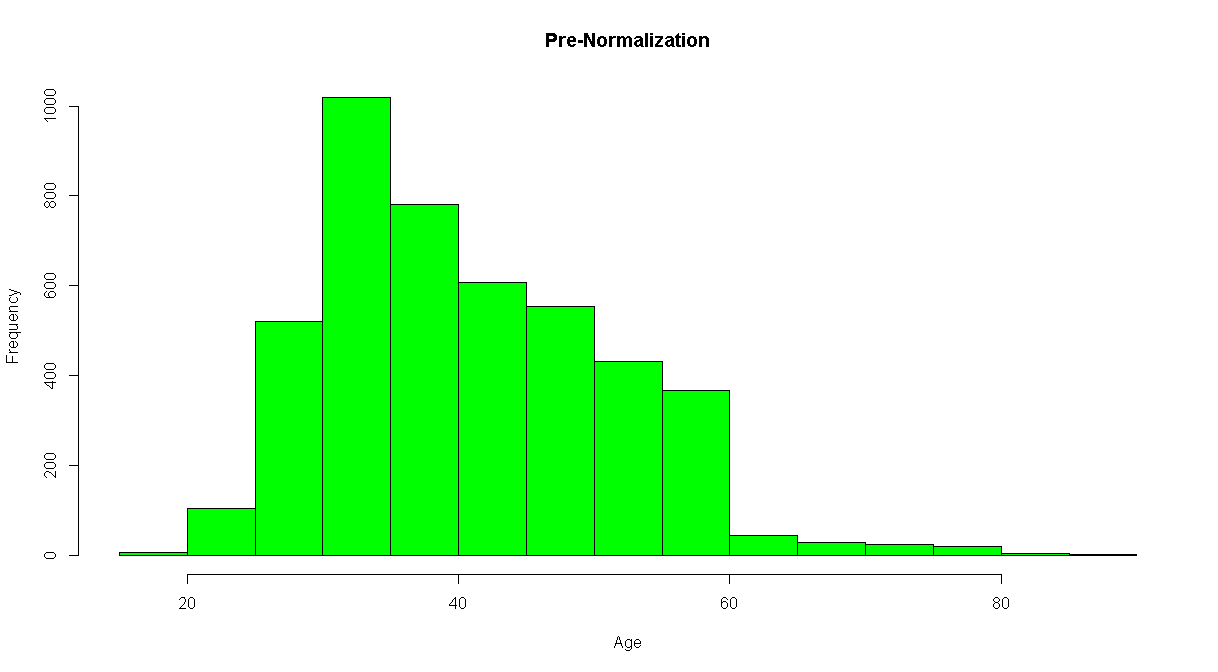


**Attribute Imbalance Summary:**

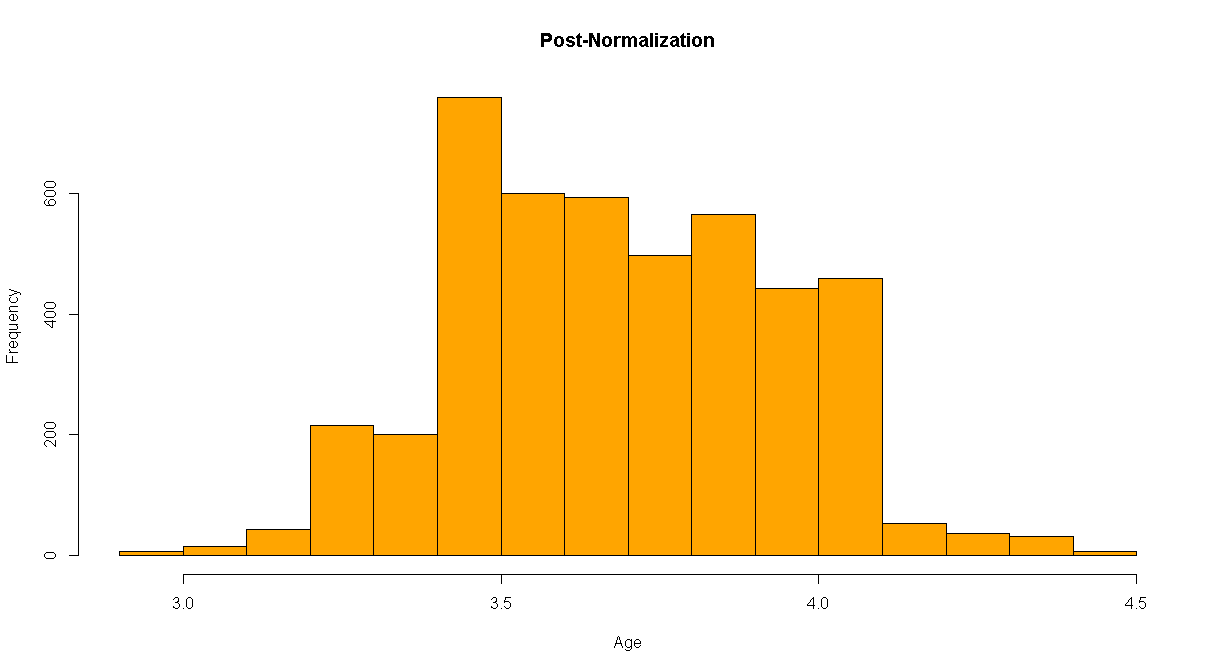
The class attributes experiences imbalance in the class attribute as there are considerable amount of No classification in the class attribute. The issue in the data becomes when we have under performing classifiers due to the imbalance in the data, also the data can be over fit because of the imbalance.

**Data Transformation:**

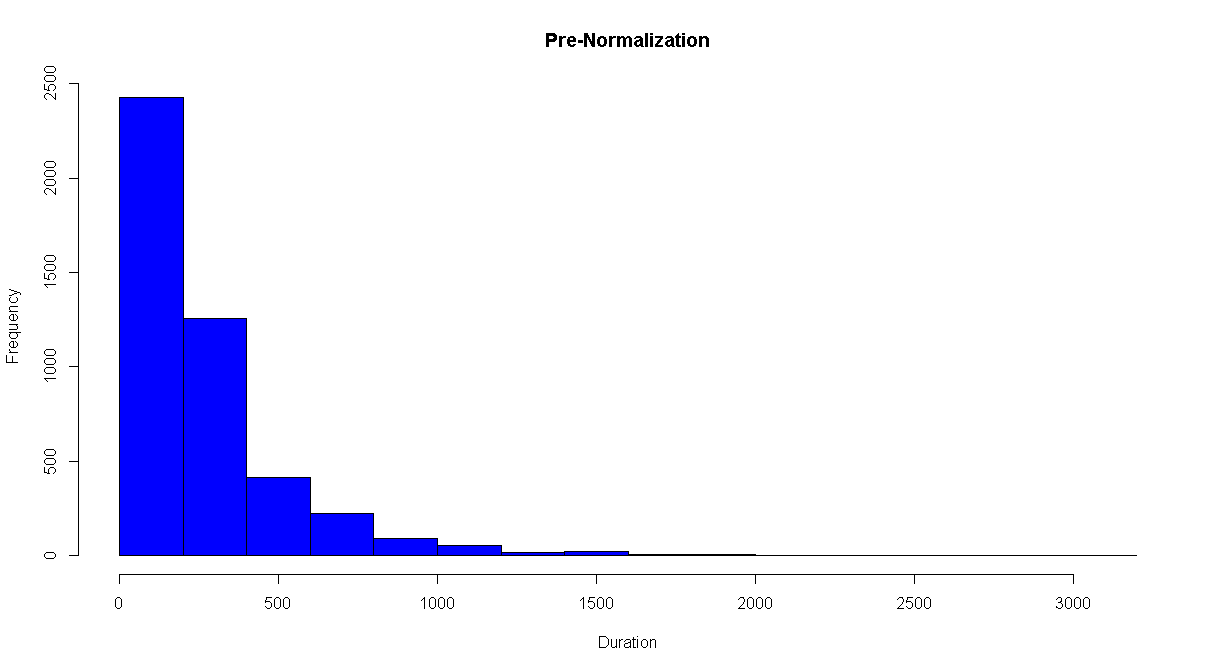
* Normalization
  + Pre Normalization (Age)



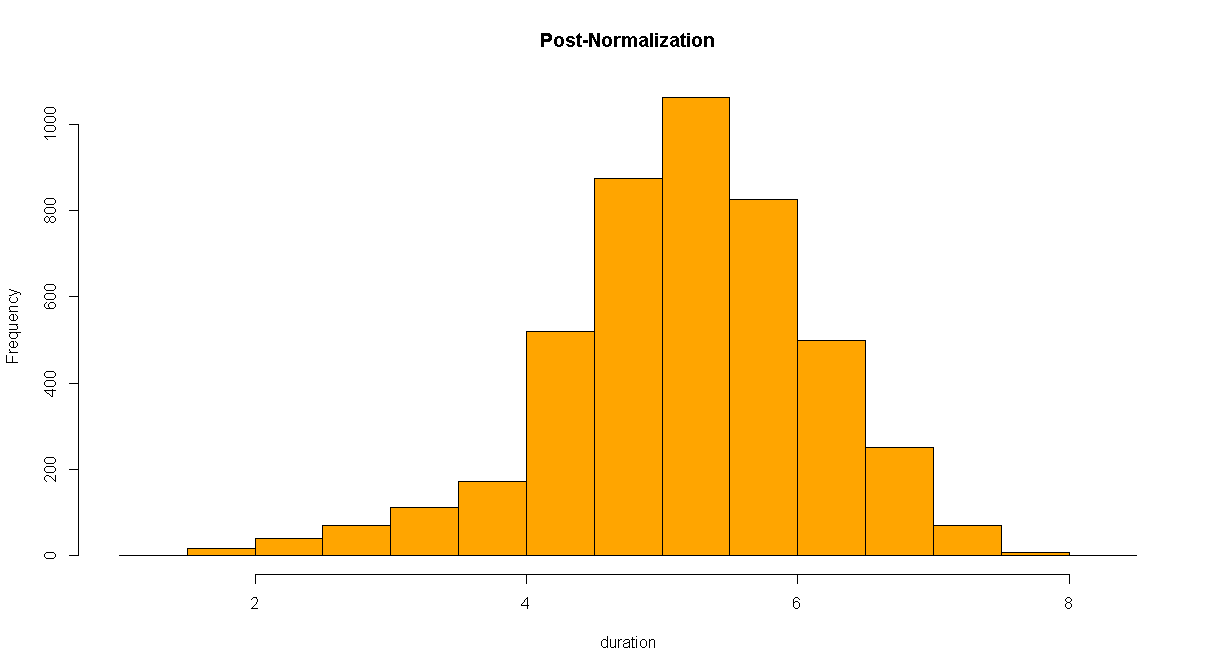
* + Post Normalization (Age)



* + Pre Normalization(Duration)



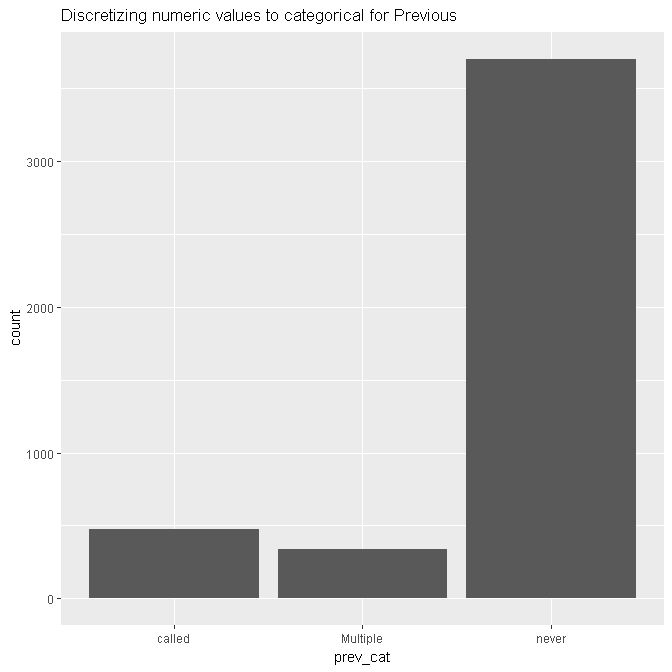
* + Post Normalization(Duration)

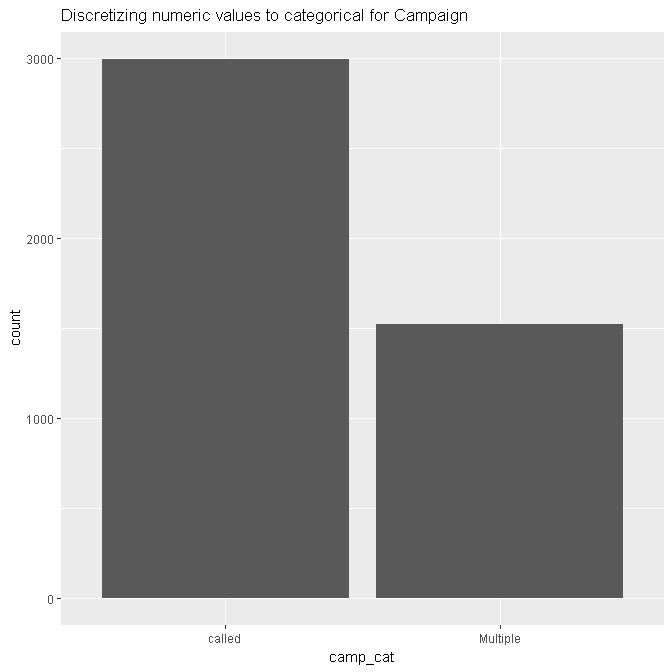


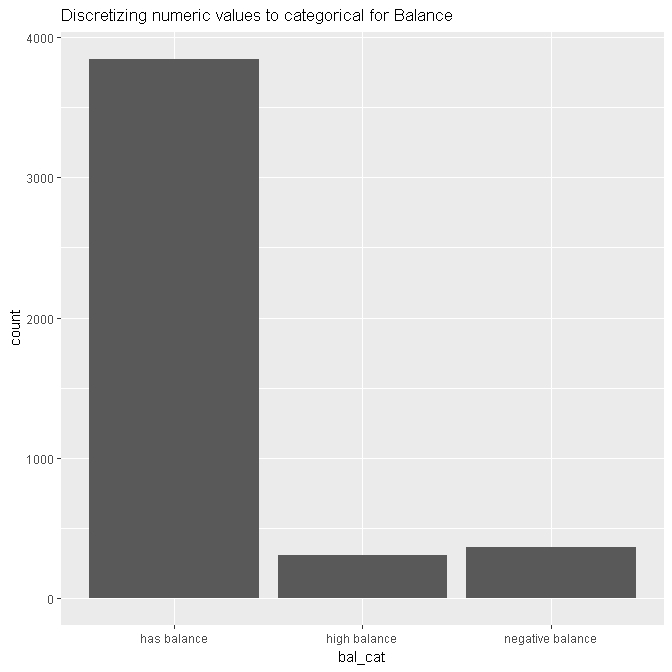
**Normalization Summary:**

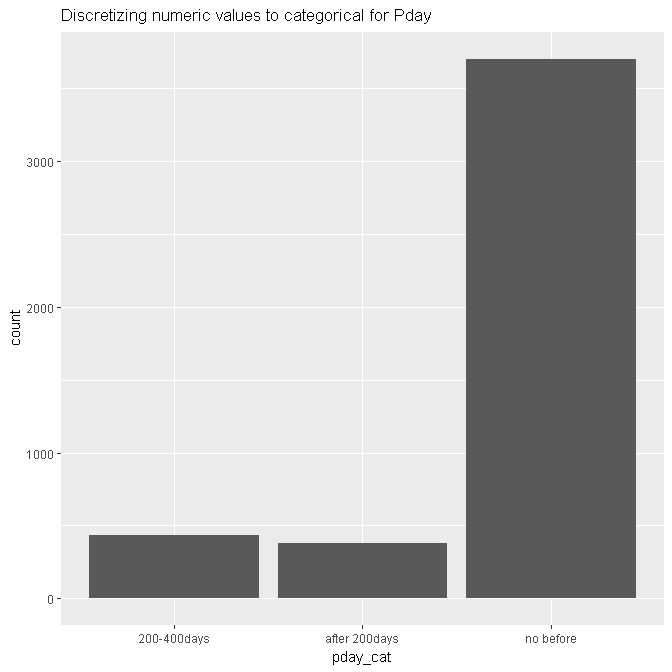
The age and the duration data had been normalized. The reason being is that as the central theorem states that the sample mean varies normally if the data is large enough. Considering the data available, it is possible that the data is not normalling varying due to the data being skewed as can be seen in the pre-Normalization bar graph. As a result the confidence interval has a wrong coverage probability due to the small sample size. The confidence interval can be corrected by normalizing the data, hence a symmetric shape in the bar chart. This can definitely affect the class attribute especially if the attributes with low confidence interval or skewed sample mean are highly correlated to the class attribute. The correlation can affect the results of the class attribute based on the distribution of the sample mean.

**Discretizing numeric values to categorical**









**Data Preparation Final Thoughts:**

The dataset of the bank is overall very clean and easy to handle. The dataset has only a few considerations to notice before performing the analysis and they include factors such as attribute normalization, data imbalance, data correlation and identifying the characteristics of the dataset such as outliers. All of the above serve as a way to better understand the data and to insure that there are no factors present in the dataset that can result in inaccurate results in the classification of the data. Data preparation serves as necessary initial step that can help determine the accuracy of the results found in the classification and the post classification stages.

**Data analysis**

## Weka

We used Weka to do our initial analysis. In the first step of our analysis we tried to choose the best attribute that contributes towards our response variable y and also to select the algorithm that would make the best prediction.

**Part 1: Choosing the important variables**

In order to choose the important variables we used three methods: Correlation, Info gain and Wrapper ranking in Weka. The results of these methods are given below. Based on these results, we choose the variables that we most common in the three methods to predict our response variable.

1. **Correlation Method**

Ranked attributes:

0.40112 12 duration

0.14472 16 poutcome

0.11862 9 contact

0.11671 15 previous

0.10468 7 housing

0.10409 14 pdays

0.07052 8 loan

0.06115 13 campaign

0.05618 3 marital

0.0531 11 month

0.04509 1 age

0.03606 4 education

0.03243 2 job

0.01791 6 balance

0.01124 10 day

0.0013 5 default

1. **Info Gain**

Ranked attributes:

0.10811967 12 duration

0.03758116 16 poutcome

0.03553361 14 pdays

0.0299014 11 month

0.01633501 9 contact

0.01622639 15 previous

0.00999086 2 job

0.00971603 1 age

0.00782731 7 housing

0.00533738 6 balance

0.0041129 8 loan

0.00304631 13 campaign

0.00297254 3 marital

0.00236554 4 education

0.00000121 5 default

0 10 day

1. **Wrapper**

Selected Attributes:

marital

balance

day

duration

poutcome

**Basis for selecting the most important attributes:**

Based on these results, we found the the overlap between theses top ranked attributes, which are marital, balance, month, contact, duration, poutcome, y.

For theses attributes, we use Decision Tree, Naive Bayes, Random Forest, bagging, and boosting to analyze our data.

**Part 2: Choosing the best algorithm**

We also used Weka experiment mode to check which method would generate the best results for this data set. The results of this test is given below:

The experiment choose Random Forest as the best performing algorithm

**>-< > < Resultset**

0 0 0 trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698

0 0 0 meta.Stacking '-X 10 -M \"trees.J48 -C 0.25 -M 2\" -S 1 -num-slots 1 -B \"lazy.IBk -K 1 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first-last\\\\\\\"\\\"\" -B \"trees.J48 -C 0.25 -M 2\"' 5134738557155845452

0 0 0 meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703

0 0 0 meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974

0 0 0 trees.J48 '-C 0.25 -M 2' -217733168393644444

## Analysis in Weka

We used the best attributes from the above method and did the analysis in Weka. The overview of results from Weka are given below. (The detailed results can be found at the appendix)

### Method 1 Random forest

Test mode: 10-fold cross-validation.

=== Summary ===

Correctly Classified Instances 4057 **89.7368** %

Incorrectly Classified Instances 464 **10.2632 %**

Kappa statistic 0.4146

K&B Relative Info Score 70696.3577 %

K&B Information Score 364.8931 bits 0.0807 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 19710.0237 bits 4.3597 bits/instance

Complexity improvement (Sf) -17379.346 bits -3.8441 bits/instance

Mean absolute error 0.1317

Root mean squared error 0.2718

Relative absolute error 64.5286 %

Root relative squared error 85.1172 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.963 0.607 0.924 0.963 0.943 0.424 **0.890** 0.981 no

0.393 0.037 0.581 0.393 0.469 0.424 0.890 0.497 yes

Weighted Avg. 0.897 0.541 0.885 0.897 0.889 0.424 0.890 0.925

=== Confusion Matrix ===

a b <-- classified as

3852 148 | a = no

316 205 | b = yes === Summary ===

**Results : Random forest generated an accuracy of 89.73 % and misclassification rate of 10.2632 %. The Roc Area was 0.890.**

### Method 2 Bagged Random Forest

Test mode: 10-fold cross-validation

=== Summary ===

Correctly Classified Instances 4062  **89.8474 %**

Incorrectly Classified Instances 459 **10.1526 %**

Kappa statistic 0.4006

K&B Relative Info Score 62767.0589 %

K&B Information Score 323.9667 bits 0.0717 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 4743.5669 bits 1.0492 bits/instance

Complexity improvement (Sf) -2412.8891 bits -0.5337 bits/instance

Mean absolute error 0.1355

Root mean squared error 0.2694

Relative absolute error 66.3967 %

Root relative squared error 84.3681 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.968 0.635 0.921 0.968 0.944 0.415 **0.896** 0.984 no

0.365 0.032 0.597 0.365 0.453 0.415 0.896 0.498 yes

Weighted Avg. 0.898 0.566 0.884 0.898 0.887 0.415 0.896 0.928

=== Confusion Matrix ===

a b <-- classified as

3872 128 | a = no

331 190 | b = yes

**Results: Bagged random forest generated an accuracy of 89.84% and misclassification error of 10.15%. Also the Roc was 0.896**

### Method 3 Bagged Decision Tree

Test mode: 10-fold cross-validation

=== Summary ===

Correctly Classified Instances 4073 **90.0907 %**

Incorrectly Classified Instances 448 9.9093 %

Kappa statistic 0.4132

K&B Relative Info Score 85171.9577 %

K&B Information Score 439.6076 bits 0.0972 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 1625.2825 bits 0.3595 bits/instance

Complexity improvement (Sf) 705.3953 bits 0.156 bits/instance

Mean absolute error 0.1373

Root mean squared error 0.2716

Relative absolute error 67.2794 %

Root relative squared error 85.0612 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.970 0.628 0.922 0.970 0.945 0.429 0.879 0.979 no

0.372 0.030 0.616 0.372 0.464 0.429 0.879 0.499 yes

Weighted Avg. 0.901 0.559 0.887 0.901 0.890 0.429 0.879 0.923

=== Confusion Matrix ===

a b <-- classified as

3879 121 | a = no

327 194 | b = yes

**Result: The bagged decision tree generated an accuracy of 90.09% and misclassification of 9.9093 % and the Roc area was 0.979**

### Method 4 Simple decision Tree

Test mode: 10-fold cross-validation

=== Summary ===

Correctly Classified Instances 4060 **89.8031 %**

Incorrectly Classified Instances 461  **10.1969 %**

Kappa statistic 0.3818

K&B Relative Info Score 52557.0532 %

K&B Information Score 271.2686 bits 0.06 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 29698.314 bits 6.569 bits/instance

Complexity improvement (Sf) -27367.6363 bits -6.0534 bits/instance

Mean absolute error 0.1529

Root mean squared error 0.2875

Relative absolute error 74.9385 %

Root relative squared error 90.023 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.971 0.662 0.918 0.971 0.944 0.401 0.769 0.941 no

0.338 0.029 0.603 0.338 0.433 0.401 0.769 0.407 yes

Weighted Avg. 0.898 0.589 0.882 0.898 0.885 0.401 0.769 0.880

=== Confusion Matrix ===

a b <-- classified as

3884 116 | a = no

345 176 | b = yes

**Results : Accuracy of 89.8031 and misclassification of 10.1969 % and Roc area of 0.941**

## Conclusion :

**The Weka experimenter choose random forest as the best performing algorithm, but if we compare the results of each and every method above, Bagged Decision tree performed the best with 90.09 % accuracy.**

**Analysis in R**

We did the same analysis on R. R gives you more flexibility in handing and analyzing the data. Also, If we refer to the data preparation part (fig X) . We can see that the distribution of our response variable y is biased towards no. So in R, we splitter the data into training and testing set such that both sets of data has equal proportions of yes and no to better train our data. The R code for the entire analysis in R can be found in appendix 2.

Summary of the R result is given below:

### Method 1 Naive Bayes: Test Results

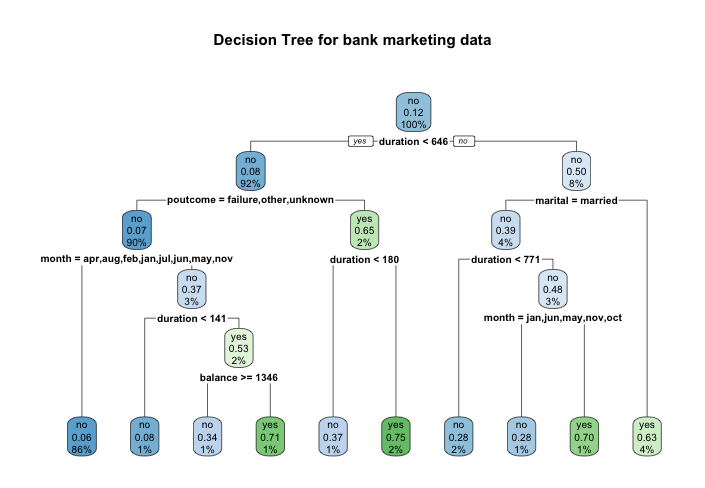
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1099 82  
## yes 101 74  
##   
## Accuracy : **0.865**    
## 95% CI : (0.8457, 0.8828)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.9891   
##   
## Kappa : 0.3706   
## Mcnemar's Test P-Value : 0.1833   
##   
## Sensitivity : **0.9158**   
## Specificity : 0.4744   
## Pos Pred Value : 0.9306   
## Neg Pred Value : 0.4229   
## Prevalence : 0.8850   
## Detection Rate : 0.8105   
## Detection Prevalence : 0.8709   
## Balanced Accuracy : 0.6951   
##   
## 'Positive' Class : no   
##

**Result: *We have a accuracy of 0.865 and sensitivity of 0.91***

### Method 2 Decision Tree: Test set result

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1158 98  
## yes 42 58  
##   
## **Accuracy : 0.8968**   
## 95% CI : (0.8793, 0.9124)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.09211   
##   
## Kappa : 0.3991   
## Mcnemar's Test P-Value : 3.346e-06   
##   
## **Sensitivity : 0.9650**   
## Specificity : 0.3718   
## Pos Pred Value : 0.9220   
## Neg Pred Value : 0.5800   
## Prevalence : 0.8850   
## Detection Rate : 0.8540   
## Detection Prevalence : 0.9263   
## Balanced Accuracy : 0.6684   
##   
## 'Positive' Class : no   
##

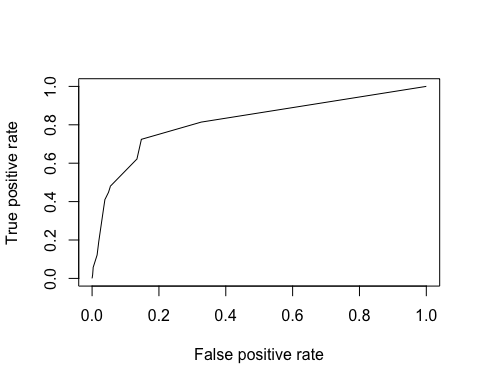
**Result: We have accuracy of 0.8968 and sensitivity of 0.9650. Also, decision tree in R generates variable importance which can be seen below:**

**DECISION TREE**

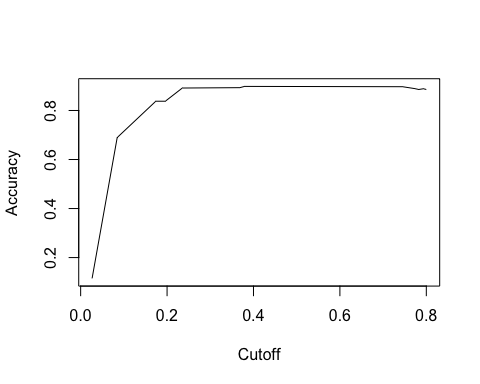
**Variable importance in Decision Tree**

**varImp**(bankmodel1)

## Overall  
## age 5.923466  
## balance 12.231944  
## campaign 5.833314  
## contact 3.336733  
## day 11.643294  
## duration 154.771557  
## job 35.983792  
## loan 3.336733  
## marital 10.835453  
## month 97.774006  
## pdays 60.005245  
## poutcome 111.365011  
## previous 48.228430  
## education 0.000000  
## default 0.000000  
## housing 0.000000

**ROC CURVE- Decision Tree**

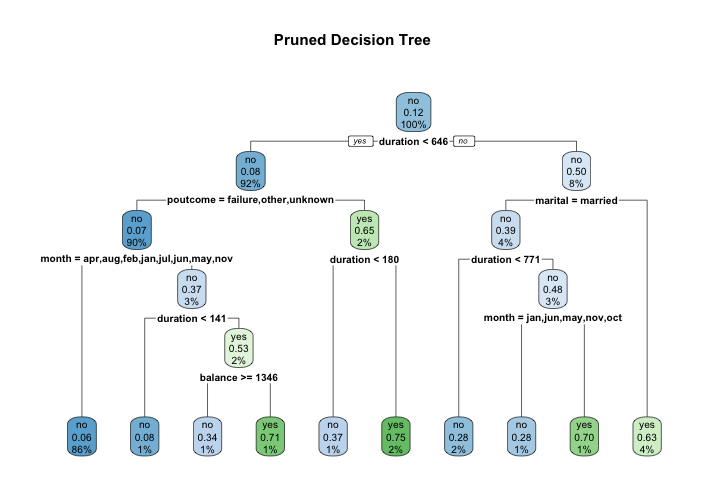
**AUC Curve - Decision Tree**



**Method 2.1 Pruning the tree ( it gives the same result as the original decision tree)**

We also tried to prune the decision tree above to improve its performance and it gave the exactly same results

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1158 98  
## yes 42 58  
##   
## Accuracy : 0.8968   
## 95% CI : (0.8793, 0.9124)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.09211   
##   
## Kappa : 0.3991   
## Mcnemar's Test P-Value : 3.346e-06   
##   
## Sensitivity : 0.9650   
## Specificity : 0.3718   
## Pos Pred Value : 0.9220   
## Neg Pred Value : 0.5800   
## Prevalence : 0.8850   
## Detection Rate : 0.8540   
## Detection Prevalence : 0.9263   
## Balanced Accuracy : 0.6684   
##   
## 'Positive' Class : no

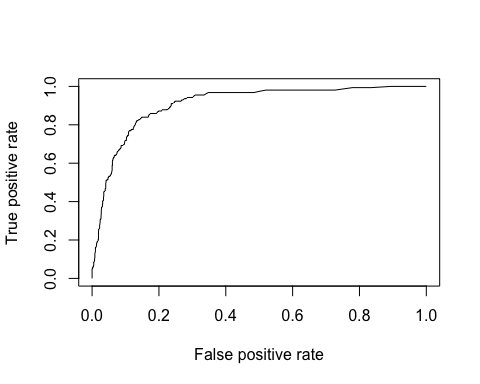


### Method 3 Random Forest

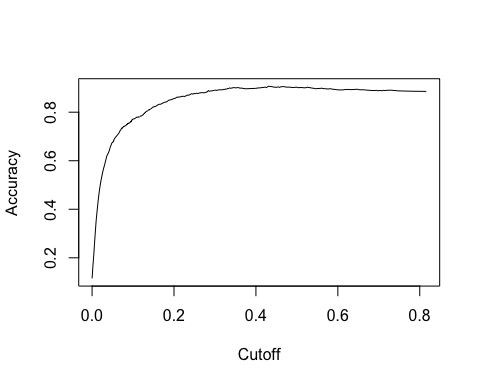
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1162 96  
## yes 38 60  
##   
## Accuracy : 0.9012   
## 95% CI : (0.884, 0.9165)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.03162   
##   
## Kappa : 0.421   
## Mcnemar's Test P-Value : 8.477e-07   
##   
## Sensitivity : 0.9683   
## Specificity : 0.3846   
## Pos Pred Value : 0.9237   
## Neg Pred Value : 0.6122   
## Prevalence : 0.8850   
## Detection Rate : 0.8569   
## Detection Prevalence : 0.9277   
## Balanced Accuracy : 0.6765   
##   
## 'Positive' Class : no   
##

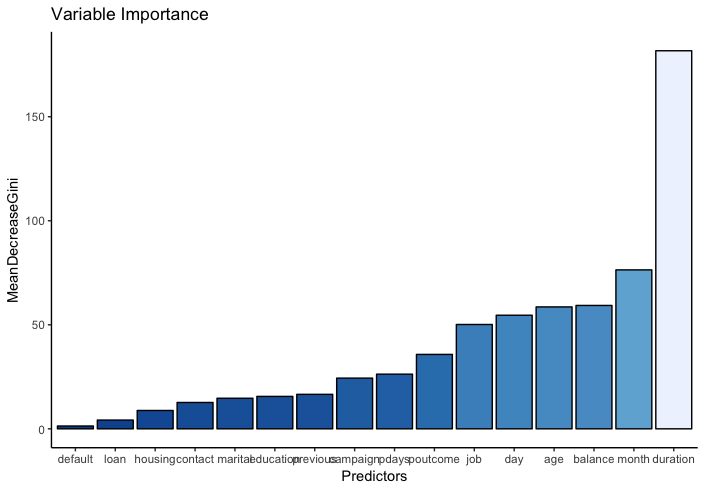
**Results: Accuracy of 0.9012 and Sensitivity of 0.9683**

**ROC CURVE**



**AUC CURVE**



**Random Forest Variable Importance:**

**Method 3.1 optimizing random forest with bagging and boosting**

**Accuracy actually went down with bagging and boosting.**

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1145 86  
## yes 55 70  
##   
## Accuracy : 0.896   
## 95% CI : (0.8785, 0.9118)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.10744   
##   
## Kappa : 0.441   
## Mcnemar's Test P-Value : 0.01152   
##   
## Sensitivity : 0.9542   
## Specificity : 0.4487   
## Pos Pred Value : 0.9301   
## Neg Pred Value : 0.5600   
## Prevalence : 0.8850   
## Detection Rate : 0.8444   
## Detection Prevalence : 0.9078   
## Balanced Accuracy : 0.7014   
##   
## 'Positive' Class : no   
##

**BOOSTING**

## var rel.inf  
## duration duration 46.6668109  
## month month 16.1321179  
## poutcome poutcome 15.8333809  
## pdays pdays 4.5932711  
## job job 4.3792518  
## day day 3.3566217  
## balance balance 2.5386102  
## age age 2.4497247  
## previous previous 1.7539019  
## marital marital 1.2222261  
## housing housing 0.3675322  
## education education 0.3442003  
## campaign campaign 0.2178374  
## loan loan 0.1281135  
## default default 0.0163995  
## contact contact 0.0000000

## Comparing results of different models from Weka and R:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Forest** | **Decision Tree** | **Bagging** | **Naive Bayes** | **Pruned Decision Tree** | **Boosting** |
| **Weka** |  |  |  |  |  |  |
| Accuracy | 89.73% | 89.80% | 90.09 | 89.32 | 89.80 | Var Imp |
| Sensitivity | 0.963 | 0.961 | 0.961 | 0.96 | 0.971 | Var Imp |
| **R** |  |  |  |  |  |  |
| Accuracy | **90.12%** | 89.68% | 89.6% | 86.5% | 89.68% | Var Imp |
| Sensitivity | **0.9683** | 0.9650 | 0.958 | 0.9158 | 0.9650 | Var Imp |

## Variable Importance: From the best selected model

**Results**

**Model Selection**

In this study, we have applied machine learning techniques to retail bank marketing data and explored how the techniques can be used to help the bank to conduct its marketing campaign:

Based on the analysis performed in R and Weka, Random forest in R gave the best results. Comparison of all the models are given in table X. Random forest had highest accuracy of 90.12 % , highest true positive rate of 0.9683 (Sensitivity) and reasonable feature selection chart.

### Feature importance from different models

The Random Forest, Boosting and Bagging were able to identify those features which have very important impacts on the response variable. These outputs supply very valuable insight to guide further marketing campaign they are given in figure x.

**Conclusion:**

If we look at the important predictors in their order of importance, as per Random Tree ( Our best algorithm ) they are : duration, month, balance, age, day, job, poutcome, pdays, campaign, previous, education, marital, contact, housing, loan and default. We know that some of these predictors are not in control of a bank as they are customer’s individual characteristic. Removing them from the important predictors leave us with duration, pdays, previous,housing, contact and loan. Among these duration is the most important predictor.

Based on the data analyzed and domain following are some of the suggestions:

* Agents should try to sell deposits for a particular clients on calls that last for longer duration of time
* Duration has positive effect on people saying “yes” because longer the conversation on the phone and more people will have interest towards term deposit. The bank should focus on the potential clients who have longer call duration and have reacted emphatically to the past campaign.
* The agents should personalize at whatever point possible . Using client’s name for example will increase response rate.
* Agent may target job category of manager as it corresponds to most yes.
* Also based on the results, the bank should try to sell in particular months for example August has the highest percentage of yes.
* Also, Bank can focus on a person’s individual characteristics such as age . For example older people has more chance of yes as compared to younger people.

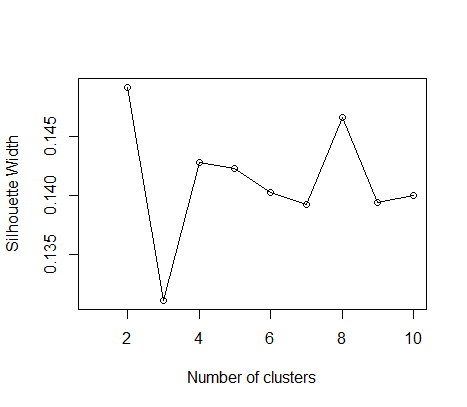
**Post Analysis**

**Clustering using R**

**Part 1 Full attributes clustering**

We used the R’s cluster package. Since the data is mixed with numerical and categorical variables. The simple K-means may not work. We use the ‘daisy’ function to calculate the ‘Gower distance’ for this dataset. Then, we used the built in function of this package to return the optimal number of clusters.

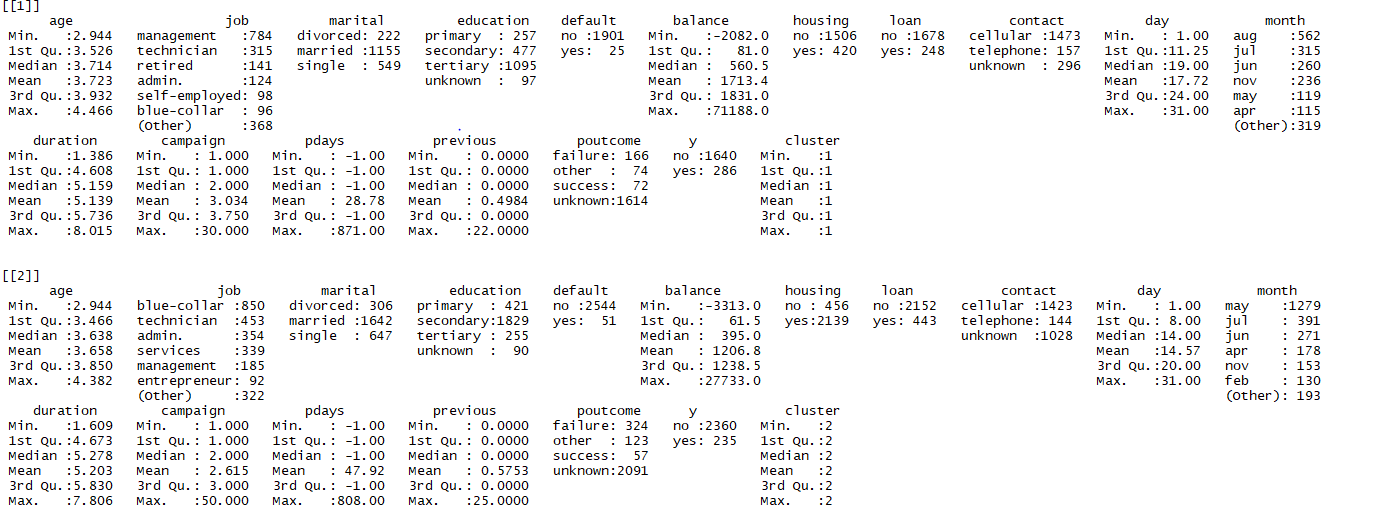
The graph is as follows:



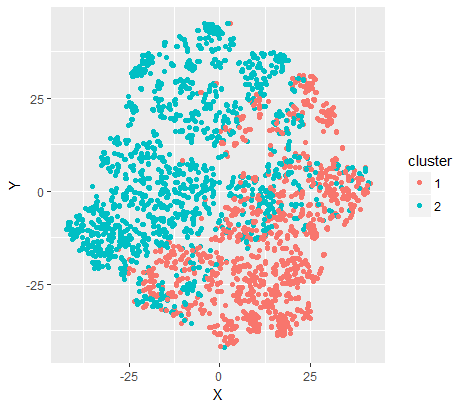
Based on the documentation, the highest Silhouette coefficient gives us the optimal number of clusters for our dataset. In this case, we can see from graph that the highest coefficient is for 2 clusters, which makes sense because we want two clusters that say yes or not to the marketing campaign.

Then, if I fit my data set into two clusters based on the Gower distance I calculated, below is the summarization of the two clusters.

We can see from the graph that in one cluster, there are 286 ‘yes’ with 1640 ‘no’, and the other cluster has 235 ‘yes’ and 2360 ‘no’. So, the first part has a relatively larger proportion of ‘yes’, which means the customers form the first cluster are more likely to be interested in the bank’s marketing campaign.

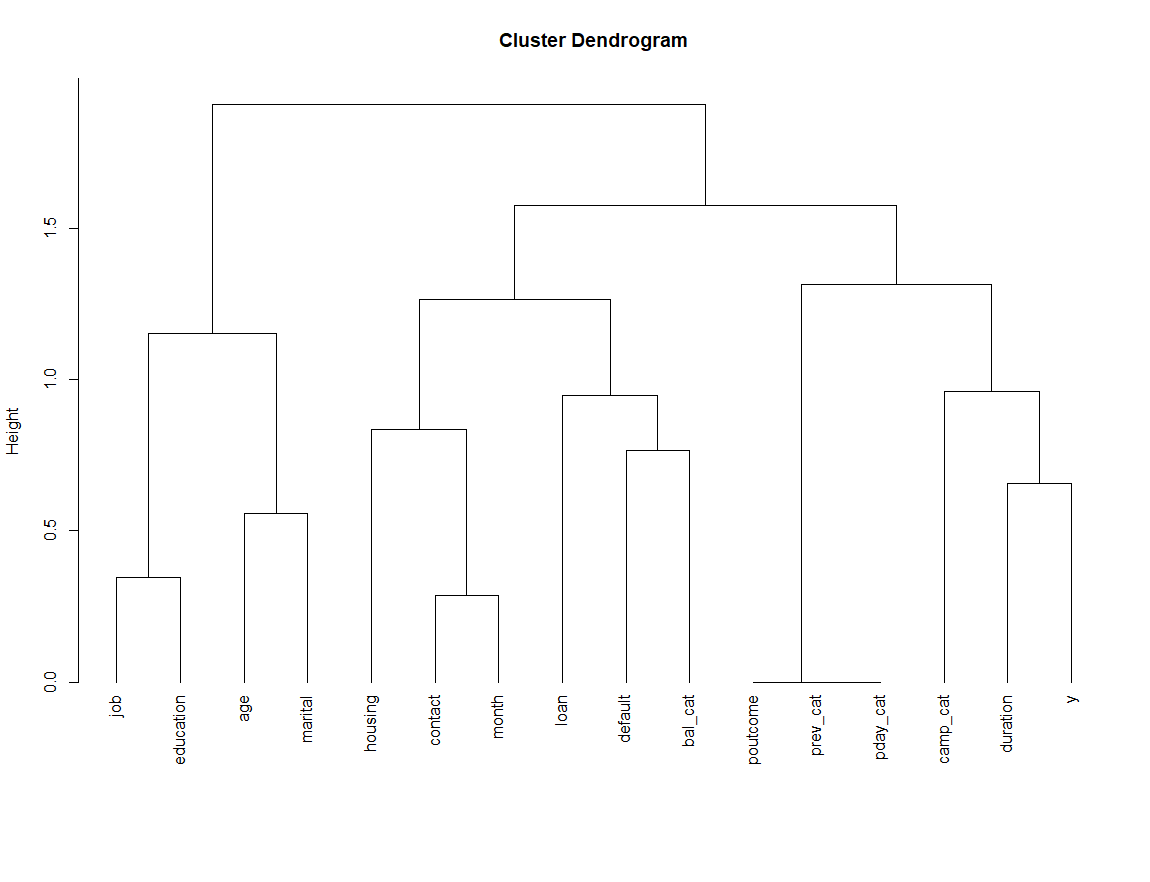


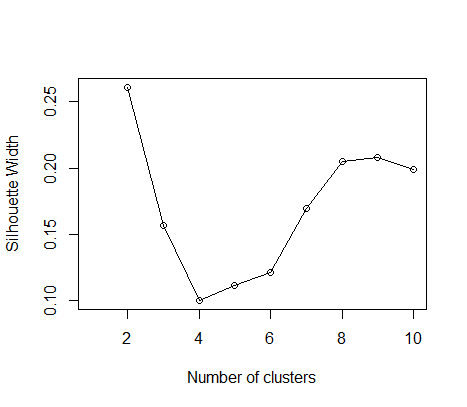
In terms of the final cluster assignment, below is the graph.



**PART 2 Clustering with selected attributes**

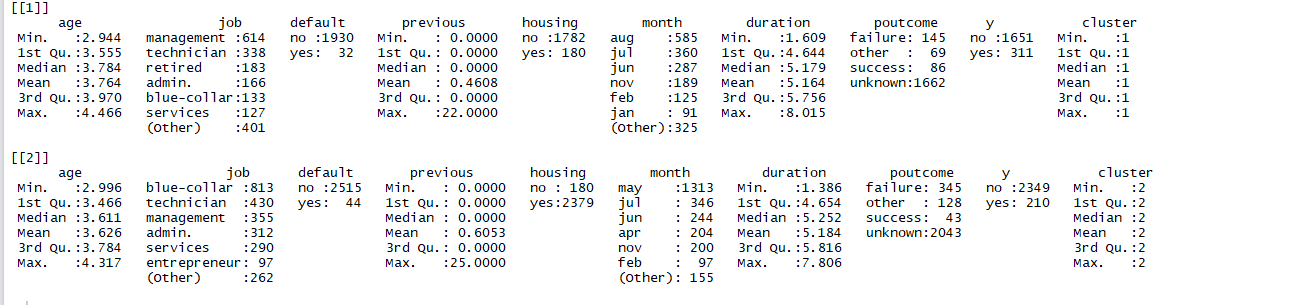
Then I did some hierarchy clustering to find the similarities between these attributes.

Here is a graph I got. For this graph, if we look from the bottom to top. For every two branches that come from one node, they are the most similar attributes. So, I did some attribute selection based on this graph. The ones I selected for clustering are age, job, default, previous, housing, month, duration, poutcome, and y.

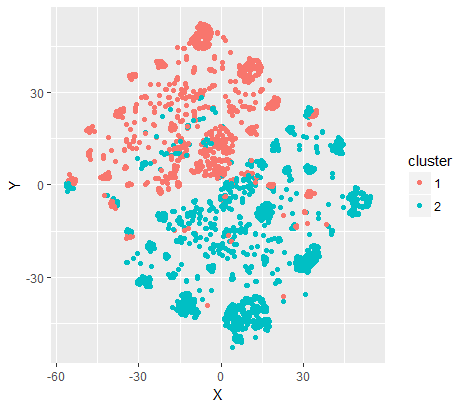
Then I used the same clustering method as mentioned in part 1. 

From the Silhouette graph, we can found that it still returns us the optimal number of clusters as 2.

And the summary of clusters is given below



Then below is the graph for the cluster assignment.



|  |  |  |
| --- | --- | --- |
| ‘Yes’ RATIO | Full attributes | Selected attributes |
| Cluster 1 | 0.174 | 0.188 |
| Cluster 2 | 0.099 | 0.089 |

Compare these two methods,

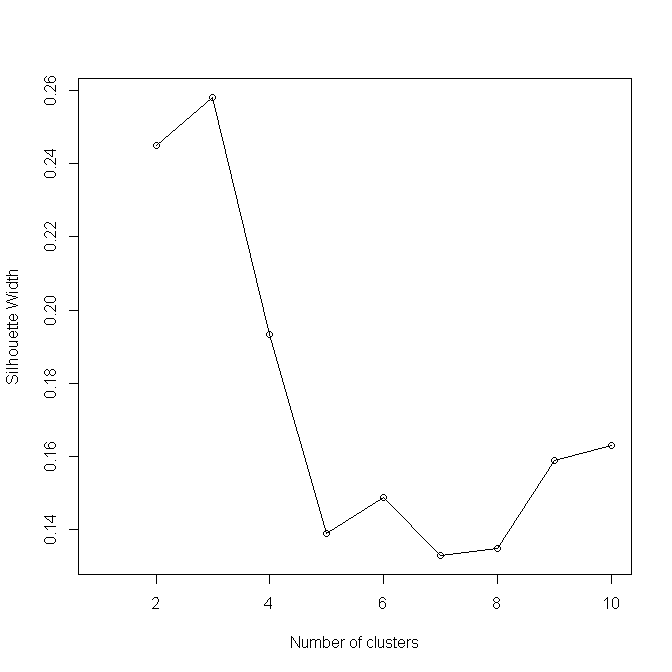
We can find that after the attributes selection, the ‘yes’ ration in the first cluster increased and the ‘no’ ratio decreased. So, by doing this selection, the cluster we got become more representative of the customers that will or will not interested in the phone marketing campaign.

**PART3 CLUSTERING WITH TRANFORMED DATA**

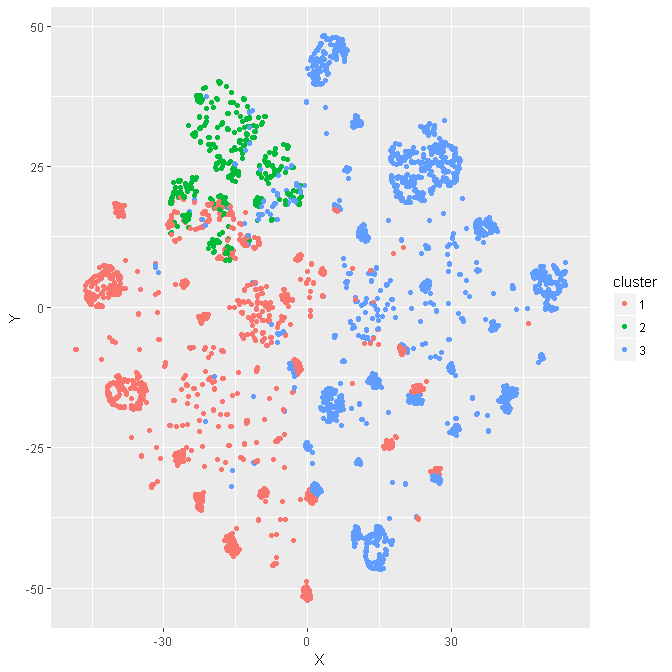
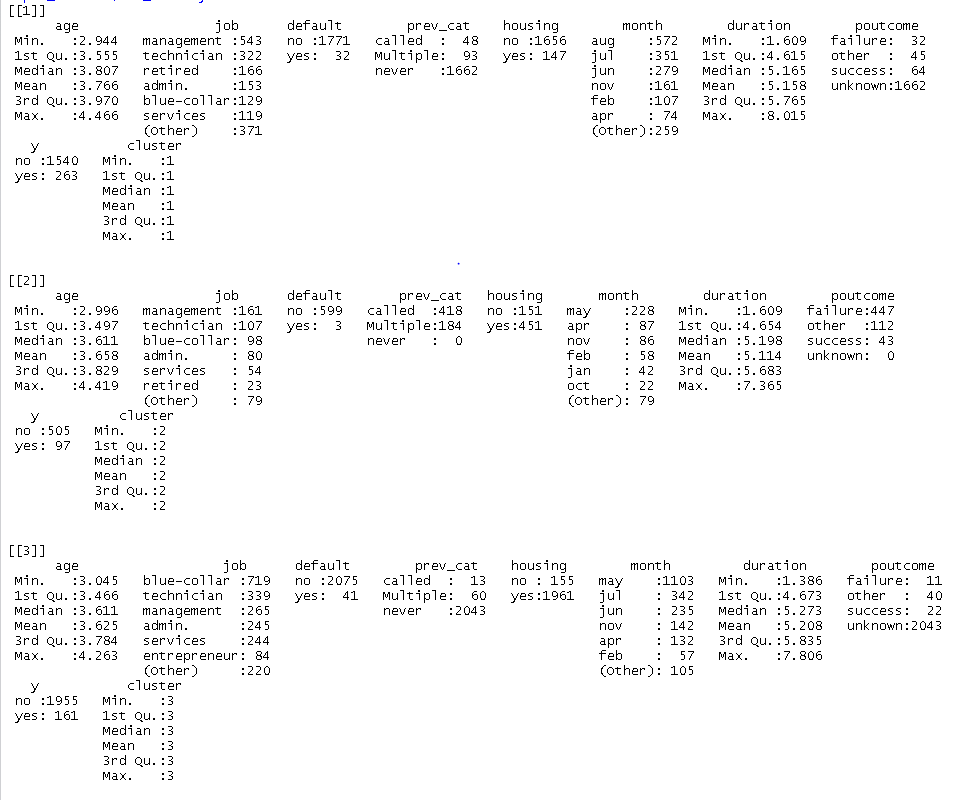
For the clustering I did above, I only normalize the age and duration data. For the other numerical attributes, as they are highly skewed, like 3500 for one small interval, we think it makes more sense to convert it to a categorical variable. If we do all these analyses after transforming the numerical variables. The optimal clusters went up to 3 for both the full attributes and selected attributes. Below is the result and summary.

**Selected Attributes**

**Silhouette coefficient Graph**

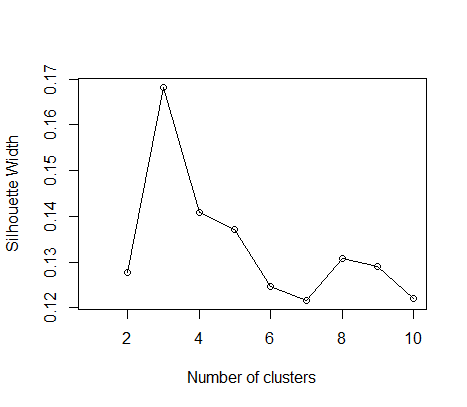


**Cluster summary**

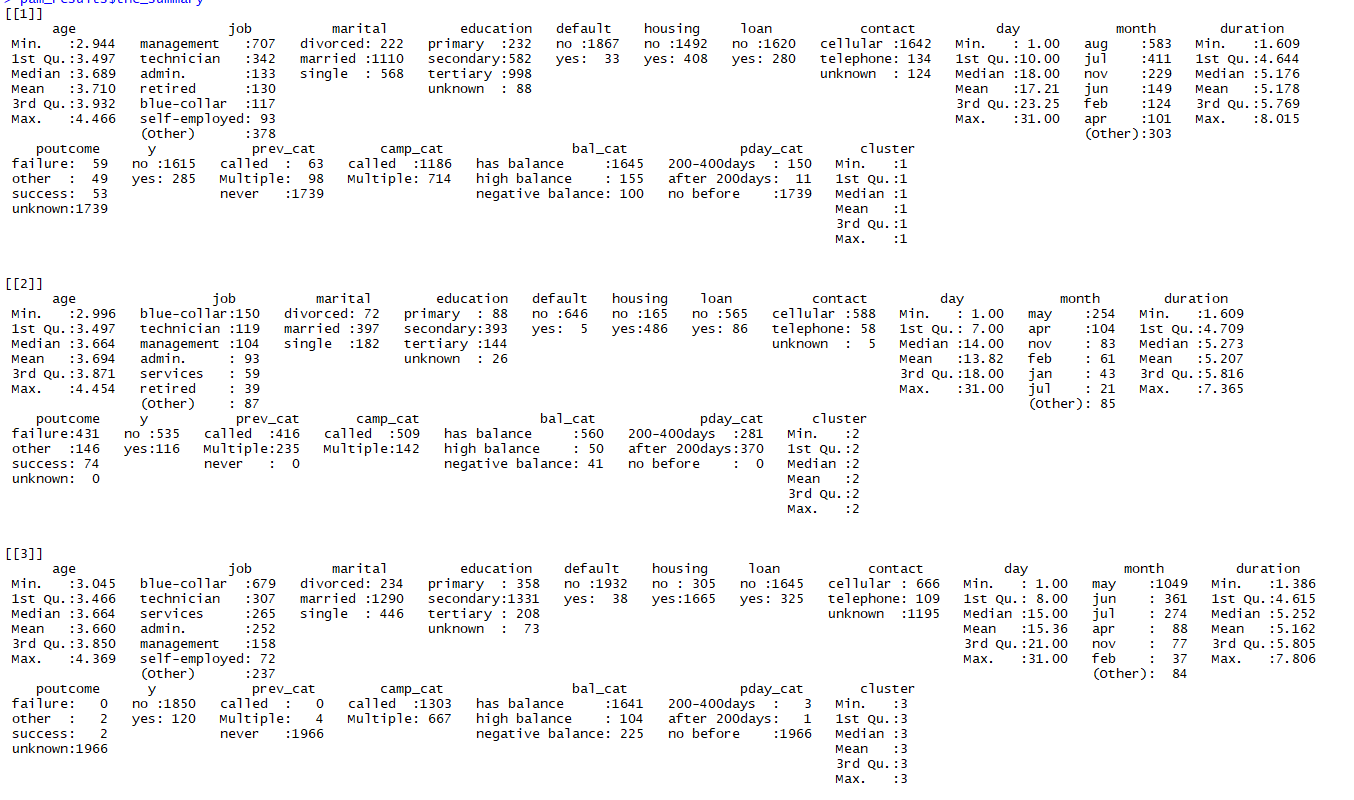


**Full attributes**

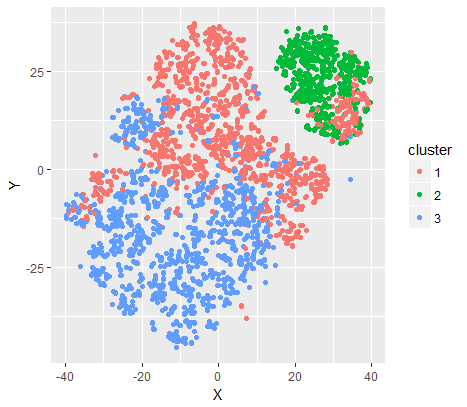
**Silhouette coefficient Graph**



**Cluster Summary**



**Cluster assignment**

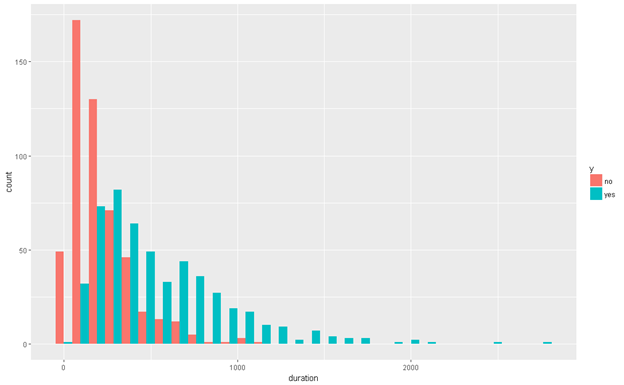
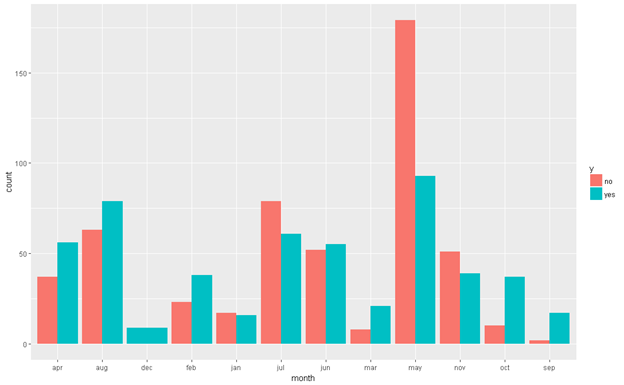
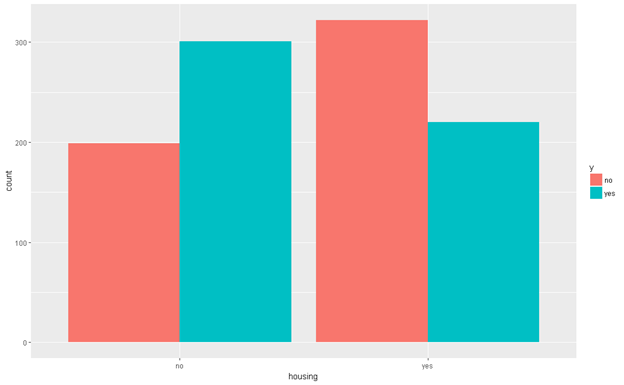
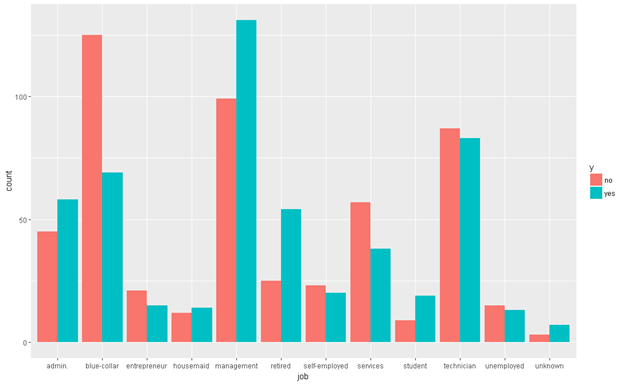
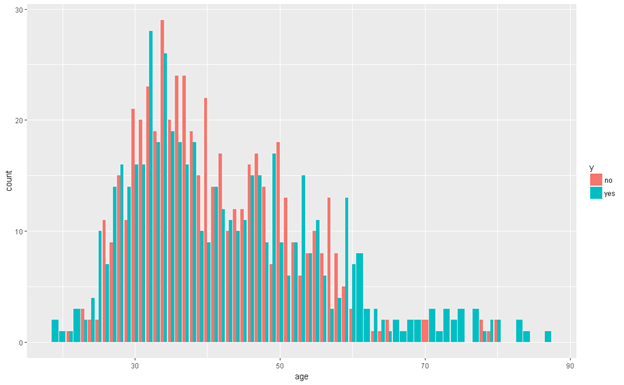


If we take a look at the clustering for the transformed the data, we can still find there is one cluster with higher ‘yes’ ratio compare two the other clusters. So, basically the original ‘no’ clusters got split into two clusters. Specifically, for the selected attributes clustering, we can find that the original ‘no’ cluster split two clusters with customers that never been called before and bank have contacted them before.

At last, in terms of what type of customers the bank should target in the future. We think they should consider the attributes that differ among these clusters. The ones we found differs among the selected attributes are age, job, housing, month, duration.

In order to show the difference, I randomly selected equal amount of ‘no’ customers and plot these attributes together with the ‘yes’ customers.

**Graph is as below:**



**Conclusion**

So, in terms of suggestions for the bank. The first part we want to mention is that background of the customers. The reason we think a customer will be interested in the long term deposit is that they are risk averse and they have extra money they don’t need to use for some time. So, for customers in higher job positions, they tend to have more extra money compare to customers in lower job positions. As we can see from the job graph, customers with more management title are most likely to say yes. Also, for the age factor, people tend to get more risk averse as they get older and they will have relatively large amount of deposit compare to the younger customers. Also, the older people tend to form a fixed money spending habit and they would like saving money for the future. On the contrary, the younger people are more risk love, they don’t tend to have a fixed money spending plan. Given that they probably don’t have a fair amount of property. it will be hard for them to deposit earnings for a long time. At last, customers saying yes also tend not to have any housing loans. It’s the similar reason, if they are making loans, they possibly don’t have the extra money to deposit.

Besides the customer selection part, what the bank should improve themselves are as follows. First thing notice is that most of the customer from ‘no’ group are contacted in May and most of the customer form ‘yes’ group are contacted in August. We don’t know the exact situation that happened. They may have different promotion plan for these two month, maybe the one that in August is more attractive. Also, maybe they use different group of people to contact the customers. And the group in August is just better than the one in May. Or they use the same group, but the group members learn from their contact experience from May and they use a better strategy. And the other factor that most directly linked with the customers’ attitude is the duration. The longer they stay on the phone, the more likely they will sign up for the deposit account.

In a word, the bank should find the right customers first, the ones that more likely to interested in long term deposit. After customer selection, they have to come up with some very good promotion plans to attract the customers. Also, they have to train the marketing team to develop better strategy to approach customers. They should try to catch customer’s attention and let the customer stay on the phone as long as possible.

**APPENDIX I - WEKA**

Corelation

=== Run information ===

Evaluator: weka.attributeSelection.CorrelationAttributeEval

Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1

Relation: bank

Instances: 4521

Attributes: 17

age

job

marital

education

default

balance

housing

loan

contact

day

month

duration

campaign

pdays

previous

poutcome

y

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:

Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 17 y):

Correlation Ranking Filter

Ranked attributes:

0.40112 12 duration

0.14472 16 poutcome

0.11862 9 contact

0.11671 15 previous

0.10468 7 housing

0.10409 14 pdays

0.07052 8 loan

0.06115 13 campaign

0.05618 3 marital

0.0531 11 month

0.04509 1 age

0.03606 4 education

0.03243 2 job

0.01791 6 balance

0.01124 10 day

0.0013 5 default

Selected attributes: 12,16,9,15,7,14,8,13,3,11,1,4,2,6,10,5 : 16

Info gain

=== Run information ===

Evaluator: weka.attributeSelection.InfoGainAttributeEval

Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1

Relation: bank

Instances: 4521

Attributes: 17

age

job

marital

education

default

balance

housing

loan

contact

day

month

duration

campaign

pdays

previous

poutcome

y

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:

Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 17 y):

Information Gain Ranking Filter

Ranked attributes:

0.10811967 12 duration

0.03758116 16 poutcome

0.03553361 14 pdays

0.0299014 11 month

0.01633501 9 contact

0.01622639 15 previous

0.00999086 2 job

0.00971603 1 age

0.00782731 7 housing

0.00533738 6 balance

0.0041129 8 loan

0.00304631 13 campaign

0.00297254 3 marital

0.00236554 4 education

0.00000121 5 default

0 10 day

Selected attributes: 12,16,14,11,9,15,2,1,7,6,8,13,3,4,5,10 : 16

WRAPPER

=== Run information ===

Evaluator: weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R 1 -E DEFAULT -- -C 0.25 -M 2

Search: weka.attributeSelection.BestFirst -D 1 -N 5

Relation: bank

Instances: 4521

Attributes: 17

age

job

marital

education

default

balance

housing

loan

contact

day

month

duration

campaign

pdays

previous

poutcome

y

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:

Best first.

Start set: no attributes

Search direction: forward

Stale search after 5 node expansions

Total number of subsets evaluated: 138

Merit of best subset found: 0.9

Attribute Subset Evaluator (supervised, Class (nominal): 17 y):

Wrapper Subset Evaluator

Learning scheme: weka.classifiers.trees.J48

Scheme options: -C 0.25 -M 2

Subset evaluation: classification accuracy

Number of folds for accuracy estimation: 5

Selected attributes: 3,6,10,12,16 : 5

marital

balance

day

duration

poutcome

EXPERIMENT – HIGHEST ACCURACY

ester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -V -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width 0 -row-name-width 25 -mean-width 0 -stddev-width 0 -sig-width 0 -count-width 5 -show-stddev -print-col-names -print-row-names -enum-col-names"

Analysing: Percent\_correct

Datasets: 1

Resultsets: 5

Confidence: 0.05 (two tailed)

Sorted by: -

Date: 6/2/17 12:09 PM

Dataset (1) trees.J48 '-C | (2) meta.AdaBoo (3) meta.Baggin (4) meta.Stacki (5) trees.Rando

--------------------------------------------------------------------------------------------------------------

bank (100) 89.38(1.10) | 89.01(1.16) 89.71(1.02) 89.44(1.05) 89.57(0.91)

--------------------------------------------------------------------------------------------------------------

(v/ /\*) | (0/1/0) (0/1/0) (0/1/0) (0/1/0)

Key:

(1) trees.J48 '-C 0.25 -M 2' -217733168393644444

(2) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974

(3) meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703

(4) meta.Stacking '-X 10 -M \"trees.J48 -C 0.25 -M 2\" -S 1 -num-slots 1 -B \"lazy.IBk -K 1 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first-last\\\\\\\"\\\"\" -B \"trees.J48 -C 0.25 -M 2\"' 5134738557155845452

(5) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698

Best performance

Tester:

weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width 0 -row-name-width 25 -mean-width 2 -stddev-width 2 -sig-width 1 -count-width 5 -print-col-names -print-row-names -enum-col-names"

Analysing: Percent\_correct

Datasets: 1

Resultsets: 5

Confidence: 0.05 (two tailed)

Sorted by: -

Date: 6/2/17 12:16 PM

>-< > < Resultset

0 0 0 trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698

0 0 0 meta.Stacking '-X 10 -M \"trees.J48 -C 0.25 -M 2\" -S 1 -num-slots 1 -B \"lazy.IBk -K 1 -W 0 -A \\\"weka.core.neighboursearch.LinearNNSearch -A \\\\\\\"weka.core.EuclideanDistance -R first-last\\\\\\\"\\\"\" -B \"trees.J48 -C 0.25 -M 2\"' 5134738557155845452

0 0 0 meta.Bagging '-P 100 -S 1 -num-slots 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -115879962237199703

0 0 0 meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.J48 -- -C 0.25 -M 2' -1178107808933117974

0 0 0 trees.J48 '-C 0.25 -M 2' -217733168393644444

RANDOM FOREST

=== Run information ===

Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: bank-weka.filters.unsupervised.attribute.Remove-R3-8,10-11,13

Instances: 4521

Attributes: 8

age

job

contact

duration

pdays

previous

poutcome

y

Test mode: split 66.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.55 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.7 seconds

=== Summary ===

Correctly Classified Instances 1367 88.9395 %

Incorrectly Classified Instances 170 11.0605 %

Kappa statistic 0.3414

K&B Relative Info Score -240.7861 %

K&B Information Score -1.2363 bits -0.0008 bits/instance

Class complexity | order 0 800.8489 bits 0.521 bits/instance

Class complexity | scheme 16667.5258 bits 10.8442 bits/instance

Complexity improvement (Sf) -15866.6769 bits -10.3231 bits/instance

Mean absolute error 0.1523

Root mean squared error 0.2975

Relative absolute error 74.3575 %

Root relative squared error 92.5267 %

Total Number of Instances 1537

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.966 0.689 0.914 0.966 0.939 0.358 0.818 0.963 no

0.311 0.034 0.549 0.311 0.397 0.358 0.818 0.396 yes

Weighted Avg. 0.889 0.612 0.871 0.889 0.876 0.358 0.818 0.897

=== Confusion Matrix ===

a b <-- classified as

1311 46 | a = no

124 56 | b = yes

BAGGED RF

=== Run information ===

Scheme: weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.RandomForest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: bank-weka.filters.unsupervised.attribute.Remove-R1-2,4-5,7-8,10,13-15

Instances: 4521

Attributes: 7

marital

balance

contact

month

duration

poutcome

y

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Bagging with 10 iterations and base learner

weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Time taken to build model: 2.52 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 4062 89.8474 %

Incorrectly Classified Instances 459 10.1526 %

Kappa statistic 0.4006

K&B Relative Info Score 62767.0589 %

K&B Information Score 323.9667 bits 0.0717 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 4743.5669 bits 1.0492 bits/instance

Complexity improvement (Sf) -2412.8891 bits -0.5337 bits/instance

Mean absolute error 0.1355

Root mean squared error 0.2694

Relative absolute error 66.3967 %

Root relative squared error 84.3681 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.968 0.635 0.921 0.968 0.944 0.415 0.896 0.984 no

0.365 0.032 0.597 0.365 0.453 0.415 0.896 0.498 yes

Weighted Avg. 0.898 0.566 0.884 0.898 0.887 0.415 0.896 0.928

=== Confusion Matrix ===

a b <-- classified as

3872 128 | a = no

331 190 | b = yes

BAGGED DT

=== Run information ===

Scheme: weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2

Relation: bank-weka.filters.unsupervised.attribute.Remove-R1-2,4-5,7-8,10,13-15

Instances: 4521

Attributes: 7

marital

balance

contact

month

duration

poutcome

y

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Bagging with 10 iterations and base learner

weka.classifiers.trees.J48 -C 0.25 -M 2

Time taken to build model: 0.13 seconds

=== Predictions on test data ===

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 4073 90.0907 %

Incorrectly Classified Instances 448 9.9093 %

Kappa statistic 0.4132

K&B Relative Info Score 85171.9577 %

K&B Information Score 439.6076 bits 0.0972 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 1625.2825 bits 0.3595 bits/instance

Complexity improvement (Sf) 705.3953 bits 0.156 bits/instance

Mean absolute error 0.1373

Root mean squared error 0.2716

Relative absolute error 67.2794 %

Root relative squared error 85.0612 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.970 0.628 0.922 0.970 0.945 0.429 0.879 0.979 no

0.372 0.030 0.616 0.372 0.464 0.429 0.879 0.499 yes

Weighted Avg. 0.901 0.559 0.887 0.901 0.890 0.429 0.879 0.923

=== Confusion Matrix ===

a b <-- classified as

3879 121 | a = no

327 194 | b = yes

SIMPLE TREE

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: bank-weka.filters.unsupervised.attribute.Remove-R1-2,4-5,7-8,10,13-15

Instances: 4521

Attributes: 7

marital

balance

contact

month

duration

poutcome

y

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

------------------

duration <= 211: no (2548.0/73.0)

duration > 211

| duration <= 645

| | poutcome = unknown

| | | contact = cellular

| | | | month = oct

| | | | | duration <= 329

| | | | | | balance <= 3043: yes (5.0/1.0)

| | | | | | balance > 3043: no (5.0)

| | | | | duration > 329: yes (6.0)

| | | | month = may

| | | | | duration <= 537: no (99.0/11.0)

| | | | | duration > 537

| | | | | | balance <= 1491: yes (4.0)

| | | | | | balance > 1491: no (2.0)

| | | | month = apr: no (75.0/18.0)

| | | | month = jun: yes (22.0/7.0)

| | | | month = feb

| | | | | duration <= 477: no (38.0/3.0)

| | | | | duration > 477

| | | | | | duration <= 555: yes (9.0/2.0)

| | | | | | duration > 555: no (4.0)

| | | | month = aug: no (160.0/21.0)

| | | | month = jan

| | | | | duration <= 394: no (25.0)

| | | | | duration > 394

| | | | | | duration <= 437: yes (4.0/1.0)

| | | | | | duration > 437: no (5.0)

| | | | month = jul: no (194.0/7.0)

| | | | month = nov: no (89.0/10.0)

| | | | month = sep: no (8.0/2.0)

| | | | month = mar

| | | | | marital = married: yes (4.0/1.0)

| | | | | marital = single: yes (5.0/1.0)

| | | | | marital = divorced: no (2.0)

| | | | month = dec

| | | | | balance <= 1343: yes (2.0)

| | | | | balance > 1343: no (2.0)

| | | contact = unknown: no (464.0/16.0)

| | | contact = telephone

| | | | balance <= 848: no (30.0/4.0)

| | | | balance > 848

| | | | | balance <= 1743: yes (8.0/2.0)

| | | | | balance > 1743: no (19.0/5.0)

| | poutcome = failure: no (174.0/35.0)

| | poutcome = other: no (74.0/22.0)

| | poutcome = success: yes (76.0/16.0)

| duration > 645

| | marital = married

| | | contact = cellular

| | | | month = oct: yes (3.0/1.0)

| | | | month = may

| | | | | duration <= 891: yes (15.0/4.0)

| | | | | duration > 891: no (4.0)

| | | | month = apr

| | | | | poutcome = unknown

| | | | | | balance <= 1567: no (7.0/1.0)

| | | | | | balance > 1567: yes (2.0)

| | | | | poutcome = failure: yes (2.0)

| | | | | poutcome = other: yes (2.0)

| | | | | poutcome = success: yes (1.0)

| | | | month = jun: no (2.0)

| | | | month = feb

| | | | | poutcome = unknown: yes (2.0)

| | | | | poutcome = failure: no (3.0/1.0)

| | | | | poutcome = other: yes (0.0)

| | | | | poutcome = success: yes (0.0)

| | | | month = aug

| | | | | poutcome = unknown: yes (19.0/7.0)

| | | | | poutcome = failure: no (2.0)

| | | | | poutcome = other: yes (0.0)

| | | | | poutcome = success: yes (0.0)

| | | | month = jan: no (5.0/1.0)

| | | | month = jul

| | | | | duration <= 796

| | | | | | balance <= 1613: no (11.0/1.0)

| | | | | | balance > 1613: yes (2.0)

| | | | | duration > 796: yes (17.0/5.0)

| | | | month = nov: no (17.0/6.0)

| | | | month = sep: yes (0.0)

| | | | month = mar: yes (1.0)

| | | | month = dec: no (2.0/1.0)

| | | contact = unknown: no (67.0/18.0)

| | | contact = telephone

| | | | duration <= 854: no (9.0/1.0)

| | | | duration > 854: yes (10.0/4.0)

| | marital = single: yes (101.0/39.0)

| | marital = divorced: yes (53.0/20.0)

Number of Leaves : 61

Size of the tree : 89

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 4060 89.8031 %

Incorrectly Classified Instances 461 10.1969 %

Kappa statistic 0.3818

K&B Relative Info Score 52557.0532 %

K&B Information Score 271.2686 bits 0.06 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 29698.314 bits 6.569 bits/instance

Complexity improvement (Sf) -27367.6363 bits -6.0534 bits/instance

Mean absolute error 0.1529

Root mean squared error 0.2875

Relative absolute error 74.9385 %

Root relative squared error 90.023 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.971 0.662 0.918 0.971 0.944 0.401 0.769 0.941 no

0.338 0.029 0.603 0.338 0.433 0.401 0.769 0.407 yes

Weighted Avg. 0.898 0.589 0.882 0.898 0.885 0.401 0.769 0.880

=== Confusion Matrix ===

a b <-- classified as

3884 116 | a = no

345 176 | b = yes

SIMPLE RANDOM FOREST

=== Run information ===

Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: bank-weka.filters.unsupervised.attribute.Remove-R1-2,4-5,7-8,10,13-15

Instances: 4521

Attributes: 7

marital

balance

contact

month

duration

poutcome

y

Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.3 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 4057 89.7368 %

Incorrectly Classified Instances 464 10.2632 %

Kappa statistic 0.4146

K&B Relative Info Score 70696.3577 %

K&B Information Score 364.8931 bits 0.0807 bits/instance

Class complexity | order 0 2330.6777 bits 0.5155 bits/instance

Class complexity | scheme 19710.0237 bits 4.3597 bits/instance

Complexity improvement (Sf) -17379.346 bits -3.8441 bits/instance

Mean absolute error 0.1317

Root mean squared error 0.2718

Relative absolute error 64.5286 %

Root relative squared error 85.1172 %

Total Number of Instances 4521

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.963 0.607 0.924 0.963 0.943 0.424 0.890 0.981 no

0.393 0.037 0.581 0.393 0.469 0.424 0.890 0.497 yes

Weighted Avg. 0.897 0.541 0.885 0.897 0.889 0.424 0.890 0.925

=== Confusion Matrix ===

a b <-- classified as

3852 148 | a = no

316 205 | b = yes

**APPENDIX II - R Markdown**

**Loading required Packages**

Loading required packages to carry out analysis on the bank market dataset.

**library**(caret)

## Warning: package 'caret' was built under R version 3.3.2

## Loading required package: lattice

## Loading required package: ggplot2

**library**(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

**library**(pROC)

## Warning: package 'pROC' was built under R version 3.3.2

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

**library**(rpart)

## Warning: package 'rpart' was built under R version 3.3.2

**library**(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.3.2

**library**(caret)  
**library**(readr)

## Warning: package 'readr' was built under R version 3.3.2

**Importing dataset**

The first task in analyzing the dataset is to import it in R. We have used to import the data in R using "Readr" package. After importing the dataset, it was in a object called bank and general exploratory analysis was carried out to look at the data. Then, we used sapply function to find class of each column in the dataset. As some of the columns were lablled as characters, we have changed the datatype to factors .

Next, we split the data into training and testing datasets. The data was split on 70:30 ratio. Since, our original dataset had unbaised number of yes and nos , we have splitted the data such that the training and testing datasets have equal proportions of yes and nos.

bank<- **read\_csv**("bank.csv")

## Parsed with column specification:  
## cols(  
## age = col\_integer(),  
## job = col\_character(),  
## marital = col\_character(),  
## education = col\_character(),  
## default = col\_character(),  
## balance = col\_integer(),  
## housing = col\_character(),  
## loan = col\_character(),  
## contact = col\_character(),  
## day = col\_integer(),  
## month = col\_character(),  
## duration = col\_integer(),  
## campaign = col\_integer(),  
## pdays = col\_integer(),  
## previous = col\_integer(),  
## poutcome = col\_character(),  
## y = col\_character()  
## )

*#Checking the data*  
**head**(bank)

## # A tibble: 6 x 17  
## age job marital education default balance housing loan  
## <int> <chr> <chr> <chr> <chr> <int> <chr> <chr>  
## 1 30 unemployed married primary no 1787 no no  
## 2 33 services married secondary no 4789 yes yes  
## 3 35 management single tertiary no 1350 yes no  
## 4 30 management married tertiary no 1476 yes yes  
## 5 59 blue-collar married secondary no 0 yes no  
## 6 35 management single tertiary no 747 no no  
## # ... with 9 more variables: contact <chr>, day <int>, month <chr>,  
## # duration <int>, campaign <int>, pdays <int>, previous <int>,  
## # poutcome <chr>, y <chr>

**sapply**(bank,class)

## age job marital education default balance   
## "integer" "character" "character" "character" "character" "integer"   
## housing loan contact day month duration   
## "character" "character" "character" "integer" "character" "integer"   
## campaign pdays previous poutcome y   
## "integer" "integer" "integer" "character" "character"

*#As some of the columns are labelled as characters, changing them to factors*  
bank$job<-**as.factor**(bank$job)  
bank$marital<-**as.factor**(bank$marital)  
bank$education<-**as.factor**(bank$education)  
bank$default<-**as.factor**(bank$default)  
bank$housing<-**as.factor**(bank$housing)  
bank$loan<-**as.factor**(bank$loan)  
bank$contact<-**as.factor**(bank$loan)  
bank$month<-**as.factor**(bank$month)  
bank$poutcome<-**as.factor**(bank$poutcome)  
bank$y<-**as.factor**(bank$y)  
bank$contact<-**as.factor**(bank$contact)  
  
*#Summarizing data*  
**summary**(bank)

## age job marital education   
## Min. :19.00 management :969 divorced: 528 primary : 678   
## 1st Qu.:33.00 blue-collar:946 married :2797 secondary:2306   
## Median :39.00 technician :768 single :1196 tertiary :1350   
## Mean :41.17 admin. :478 unknown : 187   
## 3rd Qu.:49.00 services :417   
## Max. :87.00 retired :230   
## (Other) :713   
## default balance housing loan contact   
## no :4445 Min. :-3313 no :1962 no :3830 no :3830   
## yes: 76 1st Qu.: 69 yes:2559 yes: 691 yes: 691   
## Median : 444   
## Mean : 1423   
## 3rd Qu.: 1480   
## Max. :71188   
##   
## day month duration campaign   
## Min. : 1.00 may :1398 Min. : 4 Min. : 1.000   
## 1st Qu.: 9.00 jul : 706 1st Qu.: 104 1st Qu.: 1.000   
## Median :16.00 aug : 633 Median : 185 Median : 2.000   
## Mean :15.92 jun : 531 Mean : 264 Mean : 2.794   
## 3rd Qu.:21.00 nov : 389 3rd Qu.: 329 3rd Qu.: 3.000   
## Max. :31.00 apr : 293 Max. :3025 Max. :50.000   
## (Other): 571   
## pdays previous poutcome y   
## Min. : -1.00 Min. : 0.0000 failure: 490 no :4000   
## 1st Qu.: -1.00 1st Qu.: 0.0000 other : 197 yes: 521   
## Median : -1.00 Median : 0.0000 success: 129   
## Mean : 39.77 Mean : 0.5426 unknown:3705   
## 3rd Qu.: -1.00 3rd Qu.: 0.0000   
## Max. :871.00 Max. :25.0000   
##

*#CreateDataPartition present in caret packagesplit in such a way that*  
*#training and testing data will have same ratio for target variable 70:30 split*  
intrain<-**createDataPartition**(y=bank$y,p=0.7,list = FALSE)  
training <- bank[intrain,]  
testing <- bank[-intrain,]  
**dim**(training)

## [1] 3165 17

**dim**(testing)

## [1] 1356 17

*#we can see if the imbalancing have been taken care of*  
**table**(testing$y)

##   
## no yes   
## 1200 156

**table**(training$y)

##   
## no yes   
## 2800 365

**Correlation Analysis**

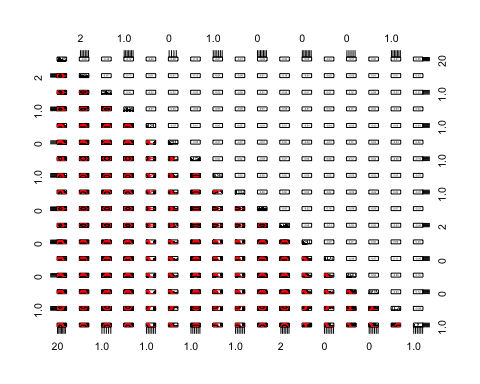
Correlation analysis can be used to tell if a predictor is a good predictor or not. This can help us to decide if some of them can be dropped. Since, the data was already cleaned , all the columns had corelation with the output yes or no

**library**(psych)

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

**pairs.panels**(bank[,])



**Analysis:**

**Model1: Naive Bayes**

*#Train model*  
**library**(e1071)

## Warning: package 'e1071' was built under R version 3.3.2

bankmodelnaive<-**naiveBayes**(y~.,data = training)  
bankmodelnaive

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## no yes   
## 0.8846761 0.1153239   
##   
## Conditional probabilities:  
## age  
## Y [,1] [,2]  
## no 41.11750 10.18714  
## yes 42.10411 13.02914  
##   
## job  
## Y admin. blue-collar entrepreneur housemaid management  
## no 0.100357143 0.213214286 0.038928571 0.023214286 0.211785714  
## yes 0.115068493 0.136986301 0.030136986 0.021917808 0.263013699  
## job  
## Y retired self-employed services student technician  
## no 0.045000000 0.044642857 0.097142857 0.017857143 0.168571429  
## yes 0.082191781 0.043835616 0.068493151 0.038356164 0.150684932  
## job  
## Y unemployed unknown  
## no 0.032857143 0.006428571  
## yes 0.035616438 0.013698630  
##   
## marital  
## Y divorced married single  
## no 0.1150000 0.6264286 0.2585714  
## yes 0.1424658 0.5205479 0.3369863  
##   
## education  
## Y primary secondary tertiary unknown  
## no 0.14714286 0.51750000 0.29428571 0.04107143  
## yes 0.11506849 0.46027397 0.37808219 0.04657534  
##   
## default  
## Y no yes  
## no 0.98535714 0.01464286  
## yes 0.97808219 0.02191781  
##   
## balance  
## Y [,1] [,2]  
## no 1419.004 3250.944  
## yes 1539.304 1990.427  
##   
## housing  
## Y no yes  
## no 0.4260714 0.5739286  
## yes 0.5671233 0.4328767  
##   
## loan  
## Y no yes  
## no 0.83500000 0.16500000  
## yes 0.91780822 0.08219178  
##   
## contact  
## Y no yes  
## no 0.83500000 0.16500000  
## yes 0.91780822 0.08219178  
##   
## day  
## Y [,1] [,2]  
## no 15.94107 8.234446  
## yes 15.44110 8.332048  
##   
## month  
## Y apr aug dec feb jan  
## no 0.058214286 0.138571429 0.003214286 0.047500000 0.029642857  
## yes 0.115068493 0.147945205 0.021917808 0.068493151 0.030136986  
## month  
## Y jul jun mar may nov  
## no 0.168928571 0.115000000 0.007142857 0.318571429 0.091785714  
## yes 0.109589041 0.128767123 0.035616438 0.178082192 0.076712329  
## month  
## Y oct sep  
## no 0.012142857 0.009285714  
## yes 0.057534247 0.030136986  
##   
## duration  
## Y [,1] [,2]  
## no 227.7757 215.4008  
## yes 568.0466 402.3812  
##   
## campaign  
## Y [,1] [,2]  
## no 2.886429 3.202908  
## yes 2.361644 2.280027  
##   
## pdays  
## Y [,1] [,2]  
## no 34.50179 92.86354  
## yes 70.29589 128.84715  
##   
## previous  
## Y [,1] [,2]  
## no 0.472500 1.671043  
## yes 1.079452 2.090465  
##   
## poutcome  
## Y failure other success unknown  
## no 0.10714286 0.03642857 0.01142857 0.84500000  
## yes 0.11506849 0.07671233 0.15068493 0.65753425

*#Make predictions with test dataset*  
predictnaive<-**predict**(bankmodelnaive,testing,type = "class")  
predictnaive

## [1] yes no no no no no yes no no yes no no no no no no no   
## [18] no no no no no no no no no no no no no no no no no   
## [35] no no no no no no no no no no yes no no yes no no no   
## [52] yes no no no yes no no no no no no no yes yes no no no   
## [69] no no no no yes no no no no no no no no no no no no   
## [86] no no no no no no no no no no yes no no yes no no no   
## [103] yes no no no no no no no yes no no no no no no no no   
## [120] no no no no no no no no no yes no yes no yes no no no   
## [137] no no no no no no no no no no no no no no no no no   
## [154] no yes no no no no no no no no no no no no no no no   
## [171] no no no no yes no no no no no no no no yes no no no   
## [188] no no no no no no yes no no no no no no no no no no   
## [205] yes no yes no yes no no no no no no no no no no no no   
## [222] yes no no no no no yes no no no no no no no no no no   
## [239] no no yes no no no no no no yes no no no no no no no   
## [256] no no no no no no no yes no no no no no no no yes no   
## [273] no no no no no no no no no yes no no no no no no no   
## [290] no no yes no no no no yes no no no no no yes no no no   
## [307] no no no no no no no no no no no no no no no no no   
## [324] no yes no no no no yes no no no no no no no no no no   
## [341] no no no no yes no no no yes no no no no no no no yes  
## [358] no yes no no no no no no no no no no yes yes no yes yes  
## [375] no yes no yes no no no no no no no no no no no no no   
## [392] no no yes no no yes no no no no no no yes no no no no   
## [409] no no no no no yes no no no yes no no no no no no yes  
## [426] no no no no no no no no no no no no no yes no yes no   
## [443] yes no no yes yes no no no no no no yes no no no no yes  
## [460] no no no no no no no no no no no no no no no no no   
## [477] no no no no no no no no yes no no yes no no no no no   
## [494] no no no no no no no no no no no yes yes no yes no no   
## [511] no no no no no no no no no yes no no no no no no yes  
## [528] no no yes no no no no no no no no no no no no no no   
## [545] no no no no no no no no no no no no yes no no no no   
## [562] no no yes no yes no no yes no no no no no yes yes no no   
## [579] yes no no no no no no yes no no yes no no no no no no   
## [596] no no no no no no no no no no no yes no no no yes no   
## [613] no no no no no no no no no no no yes no no no no yes  
## [630] yes no no no no no no no no no no no yes no no no no   
## [647] no no no no no no no no yes no no no yes no no no yes  
## [664] yes no no no no no no no no no no no no no no no no   
## [681] no no no no yes yes no no no no no no no no no no no   
## [698] no yes no yes no no no no yes no no no no no no no no   
## [715] no no no no no no no no no no no no no yes no no no   
## [732] no no no no no no no yes no no no no no no no no no   
## [749] yes no no yes no no no no no no no no no no no no no   
## [766] no no no no no yes no no no no no no no no no no no   
## [783] no no no no no no no no no yes no no no no no no no   
## [800] no no yes no no no no yes no no no no no yes no no no   
## [817] no no yes no no no no no no no yes no no yes no no no   
## [834] no no no yes no no no no no no yes no no no no no no   
## [851] no no no yes yes no no no no no no no no no no no no   
## [868] no yes no yes no no no yes no no no no no no no no no   
## [885] no no no no no no no no no no no no no no no no no   
## [902] no no no yes no no yes no no no no no no no no yes no   
## [919] no no no yes no no no no no no no no no no no no no   
## [936] no no no no no no no yes no no no no no no no no no   
## [953] no no no no no no no no no no no no yes no no yes no   
## [970] yes no no no no no no no no no no no no no no no no   
## [987] yes no no no no no no yes yes no no no no no no yes no   
## [1004] yes no no no no no yes no no no yes no no no yes no no   
## [1021] no no no no yes no no no no no yes no no yes no yes yes  
## [1038] no yes no no no no no no yes no no no no no no no no   
## [1055] yes no no no no no yes yes no no yes no no no no no yes  
## [1072] no no no yes no no no no no yes no no no yes no no no   
## [1089] no no no no no no no no no no no yes no no no no no   
## [1106] no no no yes no no no yes yes no yes no no no no no no   
## [1123] no no no no no no no no no no yes no no no no no no   
## [1140] no yes no no no no no yes no no no no no no yes no no   
## [1157] no no no no no no yes no no no yes no no no no no no   
## [1174] no no no yes no no yes yes no no no no no no yes no yes  
## [1191] yes no no no yes no no no no no no no no no no no no   
## [1208] no no no no no no no no yes no no yes no no no no no   
## [1225] yes no no no no no no no no no no no no no no no no   
## [1242] no no no no no no no no no no no no no no no no no   
## [1259] no yes no no no no no yes no no yes no no no no no no   
## [1276] yes no no yes yes yes no no no no no no no no no no yes  
## [1293] no yes no no no no no no no no no no no no no yes no   
## [1310] no no no no no no yes no no no no no no no no no no   
## [1327] no no no no no no no no no no no no no no no no no   
## [1344] yes no no no no no no no no yes no no no   
## Levels: no yes

*#checking accuracy of test dataset*  
**confusionMatrix**(predictnaive,testing$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1099 82  
## yes 101 74  
##   
## Accuracy : 0.865   
## 95% CI : (0.8457, 0.8828)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.9891   
##   
## Kappa : 0.3706   
## Mcnemar's Test P-Value : 0.1833   
##   
## Sensitivity : 0.9158   
## Specificity : 0.4744   
## Pos Pred Value : 0.9306   
## Neg Pred Value : 0.4229   
## Prevalence : 0.8850   
## Detection Rate : 0.8105   
## Detection Prevalence : 0.8709   
## Balanced Accuracy : 0.6951   
##   
## 'Positive' Class : no   
##

*# We have a accuracy of 0.8783*  
naiveaccuracy<-0.8783

**Model2: Decision Tree**

Now, we try out Decision Tree

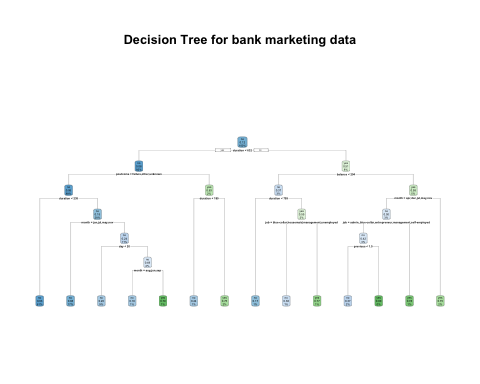
Using a decision tree improved out accuracy of prediction on training dataset to 89.75 It automatically selects the variables that contribute the most to the output variable y. We can see that in the decision tree diagram and also we can see the ranking of variables by their ranking.

The next step , we tried to prune the tree to see if there is a better model that can be made, but it is identical to the first decision tree model with same accuracy and tree.

*#Model1: Decision Tree*  
bankmodel1<-**rpart**(y~.,data=training)  
**summary**(bankmodel1)

## Call:  
## rpart(formula = y ~ ., data = training)  
## n= 3165   
##   
## CP nsplit rel error xerror xstd  
## 1 0.04383562 0 1.0000000 1.0000000 0.04923180  
## 2 0.02465753 3 0.8684932 0.9287671 0.04766573  
## 3 0.01917808 4 0.8438356 0.9150685 0.04735474  
## 4 0.01643836 6 0.8054795 0.9041096 0.04710357  
## 5 0.01164384 9 0.7561644 0.8904110 0.04678656  
## 6 0.01000000 13 0.7095890 0.8794521 0.04653049  
##   
## Variable importance  
## duration poutcome month day job balance campaign housing   
## 48 19 11 6 5 3 2 2   
## previous age pdays   
## 1 1 1   
##   
## Node number 1: 3165 observations, complexity param=0.04383562  
## predicted class=no expected loss=0.1153239 P(node) =1  
## class counts: 2800 365  
## probabilities: 0.885 0.115   
## left son=2 (2900 obs) right son=3 (265 obs)  
## Primary splits:  
## duration < 631.5 to the left, improve=88.13126, (0 missing)  
## poutcome splits as LLRL, improve=47.79696, (0 missing)  
## month splits as LLRLLLLRLLRR, improve=19.77922, (0 missing)  
## pdays < 22.5 to the left, improve=16.40348, (0 missing)  
## previous < 0.5 to the left, improve=15.92266, (0 missing)  
##   
## Node number 2: 2900 observations, complexity param=0.04383562  
## predicted class=no expected loss=0.07965517 P(node) =0.9162717  
## class counts: 2669 231  
## probabilities: 0.920 0.080   
## left son=4 (2824 obs) right son=5 (76 obs)  
## Primary splits:  
## poutcome splits as LLRL, improve=47.54831, (0 missing)  
## duration < 222.5 to the left, improve=26.63917, (0 missing)  
## month splits as LLRLLLLRLLRR, improve=22.72917, (0 missing)  
## pdays < 22.5 to the left, improve=15.80047, (0 missing)  
## previous < 0.5 to the left, improve=15.53748, (0 missing)  
##   
## Node number 3: 265 observations, complexity param=0.04383562  
## predicted class=yes expected loss=0.4943396 P(node) =0.08372828  
## class counts: 131 134  
## probabilities: 0.494 0.506   
## left son=6 (97 obs) right son=7 (168 obs)  
## Primary splits:  
## balance < 294.5 to the left, improve=5.538002, (0 missing)  
## month splits as RRLRLLRRLLL-, improve=4.696022, (0 missing)  
## duration < 759.5 to the left, improve=4.231447, (0 missing)  
## poutcome splits as LRLL, improve=4.031657, (0 missing)  
## marital splits as RLR, improve=3.639516, (0 missing)  
## Surrogate splits:  
## age < 27.5 to the left, agree=0.649, adj=0.041, (0 split)  
## job splits as RRRLRRRRRRRR, agree=0.649, adj=0.041, (0 split)  
## month splits as RRRRRLRLRRL-, agree=0.649, adj=0.041, (0 split)  
## day < 2.5 to the left, agree=0.645, adj=0.031, (0 split)  
## duration < 639.5 to the left, agree=0.642, adj=0.021, (0 split)  
##   
## Node number 4: 2824 observations, complexity param=0.01164384  
## predicted class=no expected loss=0.0648017 P(node) =0.8922591  
## class counts: 2641 183  
## probabilities: 0.935 0.065   
## left son=8 (1927 obs) right son=9 (897 obs)  
## Primary splits:  
## duration < 238.5 to the left, improve=17.831610, (0 missing)  
## month splits as RLRLLLLRLLRR, improve=12.270380, (0 missing)  
## pdays < 374.5 to the left, improve=10.113180, (0 missing)  
## age < 60.5 to the left, improve= 4.779675, (0 missing)  
## job splits as RLLLRRRLRLLR, improve= 3.281514, (0 missing)  
## Surrogate splits:  
## pdays < 414.5 to the left, agree=0.683, adj=0.002, (0 split)  
##   
## Node number 5: 76 observations, complexity param=0.02465753  
## predicted class=yes expected loss=0.3684211 P(node) =0.02401264  
## class counts: 28 48  
## probabilities: 0.368 0.632   
## left son=10 (17 obs) right son=11 (59 obs)  
## Primary splits:  
## duration < 180.5 to the left, improve=6.877893, (0 missing)  
## month splits as LLRLRRRRLRLR, improve=4.272094, (0 missing)  
## job splits as LL-LRR-LRLRL, improve=3.603197, (0 missing)  
## pdays < 93 to the right, improve=1.920145, (0 missing)  
## day < 5.5 to the right, improve=1.746301, (0 missing)  
## Surrogate splits:  
## balance < 35.5 to the left, agree=0.789, adj=0.059, (0 split)  
##   
## Node number 6: 97 observations, complexity param=0.01643836  
## predicted class=no expected loss=0.371134 P(node) =0.03064771  
## class counts: 61 36  
## probabilities: 0.629 0.371   
## left son=12 (46 obs) right son=13 (51 obs)  
## Primary splits:  
## duration < 759 to the left, improve=6.806057, (0 missing)  
## pdays < 167 to the left, improve=1.776763, (0 missing)  
## day < 25.5 to the right, improve=1.562174, (0 missing)  
## month splits as RR-RLRRRRLR-, improve=1.341482, (0 missing)  
## job splits as LLRLLRRRRRL-, improve=1.293502, (0 missing)  
## Surrogate splits:  
## job splits as LRLLRRRRRRR-, agree=0.619, adj=0.196, (0 split)  
## housing splits as LR, agree=0.588, adj=0.130, (0 split)  
## age < 44 to the left, agree=0.567, adj=0.087, (0 split)  
## day < 16.5 to the right, agree=0.567, adj=0.087, (0 split)  
## education splits as RLRR, agree=0.557, adj=0.065, (0 split)  
##   
## Node number 7: 168 observations, complexity param=0.01917808  
## predicted class=yes expected loss=0.4166667 P(node) =0.05308057  
## class counts: 70 98  
## probabilities: 0.417 0.583   
## left son=14 (109 obs) right son=15 (59 obs)  
## Primary splits:  
## month splits as LRLRRLR-LL--, improve=4.798372, (0 missing)  
## job splits as LLLRLRLRLLRR, improve=4.639086, (0 missing)  
## marital splits as RLR, improve=4.115646, (0 missing)  
## previous < 1.5 to the left, improve=3.598639, (0 missing)  
## campaign < 3.5 to the left, improve=1.840787, (0 missing)  
## Surrogate splits:  
## housing splits as RL, agree=0.750, adj=0.288, (0 split)  
## day < 6.5 to the right, agree=0.714, adj=0.186, (0 split)  
## age < 54.5 to the left, agree=0.702, adj=0.153, (0 split)  
## job splits as LLLRLRLRRRLL, agree=0.702, adj=0.153, (0 split)  
## duration < 1544.5 to the left, agree=0.667, adj=0.051, (0 split)  
##   
## Node number 8: 1927 observations  
## predicted class=no expected loss=0.02646601 P(node) =0.6088468  
## class counts: 1876 51  
## probabilities: 0.974 0.026   
##   
## Node number 9: 897 observations, complexity param=0.01164384  
## predicted class=no expected loss=0.1471572 P(node) =0.2834123  
## class counts: 765 132  
## probabilities: 0.853 0.147   
## left son=18 (544 obs) right son=19 (353 obs)  
## Primary splits:  
## month splits as RRRRLLRRLLRR, improve=10.833600, (0 missing)  
## pdays < 371.5 to the left, improve=10.003350, (0 missing)  
## job splits as RLLLRRRLRLRR, improve= 5.918489, (0 missing)  
## previous < 1.5 to the left, improve= 5.540756, (0 missing)  
## poutcome splits as LR-L, improve= 4.336689, (0 missing)  
## Surrogate splits:  
## day < 6.5 to the right, agree=0.672, adj=0.167, (0 split)  
## housing splits as RL, agree=0.656, adj=0.125, (0 split)  
## age < 59.5 to the left, agree=0.620, adj=0.034, (0 split)  
## campaign < 7.5 to the left, agree=0.614, adj=0.020, (0 split)  
## poutcome splits as LR-L, agree=0.613, adj=0.017, (0 split)  
##   
## Node number 10: 17 observations  
## predicted class=no expected loss=0.2352941 P(node) =0.005371248  
## class counts: 13 4  
## probabilities: 0.765 0.235   
##   
## Node number 11: 59 observations  
## predicted class=yes expected loss=0.2542373 P(node) =0.01864139  
## class counts: 15 44  
## probabilities: 0.254 0.746   
##   
## Node number 12: 46 observations  
## predicted class=no expected loss=0.173913 P(node) =0.01453397  
## class counts: 38 8  
## probabilities: 0.826 0.174   
##   
## Node number 13: 51 observations, complexity param=0.01643836  
## predicted class=yes expected loss=0.4509804 P(node) =0.01611374  
## class counts: 23 28  
## probabilities: 0.451 0.549   
## left son=26 (29 obs) right son=27 (22 obs)  
## Primary splits:  
## job splits as RLRLLRRRRRL-, improve=3.872457, (0 missing)  
## duration < 987 to the left, improve=1.165671, (0 missing)  
## age < 30.5 to the right, improve=1.143791, (0 missing)  
## loan splits as RL, improve=1.125032, (0 missing)  
## contact splits as RL, improve=1.125032, (0 missing)  
## Surrogate splits:  
## month splits as LR-RLRLLLLR-, agree=0.725, adj=0.364, (0 split)  
## age < 48 to the left, agree=0.647, adj=0.182, (0 split)  
## education splits as LRLR, agree=0.647, adj=0.182, (0 split)  
## housing splits as RL, agree=0.627, adj=0.136, (0 split)  
## duration < 985 to the left, agree=0.627, adj=0.136, (0 split)  
##   
## Node number 14: 109 observations, complexity param=0.01917808  
## predicted class=no expected loss=0.4954128 P(node) =0.03443918  
## class counts: 55 54  
## probabilities: 0.505 0.495   
## left son=28 (86 obs) right son=29 (23 obs)  
## Primary splits:  
## job splits as LLLRLRLR-RRR, improve=4.808861, (0 missing)  
## previous < 1.5 to the left, improve=4.336845, (0 missing)  
## marital splits as RLR, improve=3.080291, (0 missing)  
## poutcome splits as LRRL, improve=2.707443, (0 missing)  
## pdays < 86 to the left, improve=2.050469, (0 missing)  
## Surrogate splits:  
## age < 58.5 to the left, agree=0.807, adj=0.087, (0 split)  
## balance < 6983 to the left, agree=0.807, adj=0.087, (0 split)  
## day < 29.5 to the left, agree=0.807, adj=0.087, (0 split)  
## pdays < 356.5 to the left, agree=0.798, adj=0.043, (0 split)  
##   
## Node number 15: 59 observations  
## predicted class=yes expected loss=0.2542373 P(node) =0.01864139  
## class counts: 15 44  
## probabilities: 0.254 0.746   
##   
## Node number 18: 544 observations  
## predicted class=no expected loss=0.08455882 P(node) =0.1718799  
## class counts: 498 46  
## probabilities: 0.915 0.085   
##   
## Node number 19: 353 observations, complexity param=0.01164384  
## predicted class=no expected loss=0.2436261 P(node) =0.1115324  
## class counts: 267 86  
## probabilities: 0.756 0.244   
## left son=38 (296 obs) right son=39 (57 obs)  
## Primary splits:  
## day < 20.5 to the left, improve=8.334819, (0 missing)  
## month splits as LLRL--LL--RL, improve=6.293965, (0 missing)  
## job splits as RLLLRRRLRLRR, improve=4.435458, (0 missing)  
## balance < 1340 to the left, improve=3.313826, (0 missing)  
## poutcome splits as LR-L, improve=3.006561, (0 missing)  
##   
## Node number 26: 29 observations  
## predicted class=no expected loss=0.3793103 P(node) =0.009162717  
## class counts: 18 11  
## probabilities: 0.621 0.379   
##   
## Node number 27: 22 observations  
## predicted class=yes expected loss=0.2272727 P(node) =0.006951027  
## class counts: 5 17  
## probabilities: 0.227 0.773   
##   
## Node number 28: 86 observations, complexity param=0.01643836  
## predicted class=no expected loss=0.4186047 P(node) =0.0271722  
## class counts: 50 36  
## probabilities: 0.581 0.419   
## left son=56 (76 obs) right son=57 (10 obs)  
## Primary splits:  
## previous < 1.5 to the left, improve=3.292044, (0 missing)  
## loan splits as LR, improve=2.211701, (0 missing)  
## contact splits as LR, improve=2.211701, (0 missing)  
## pdays < 155.5 to the left, improve=1.937388, (0 missing)  
## poutcome splits as LRRL, improve=1.937388, (0 missing)  
## Surrogate splits:  
## pdays < 42 to the left, agree=0.965, adj=0.7, (0 split)  
## poutcome splits as RRRL, agree=0.965, adj=0.7, (0 split)  
## balance < 321.5 to the right, agree=0.895, adj=0.1, (0 split)  
##   
## Node number 29: 23 observations  
## predicted class=yes expected loss=0.2173913 P(node) =0.007266983  
## class counts: 5 18  
## probabilities: 0.217 0.783   
##   
## Node number 38: 296 observations  
## predicted class=no expected loss=0.1959459 P(node) =0.09352291  
## class counts: 238 58  
## probabilities: 0.804 0.196   
##   
## Node number 39: 57 observations, complexity param=0.01164384  
## predicted class=no expected loss=0.4912281 P(node) =0.01800948  
## class counts: 29 28  
## probabilities: 0.509 0.491   
## left son=78 (28 obs) right son=79 (29 obs)  
## Primary splits:  
## month splits as RL-R--L---RL, improve=10.759700, (0 missing)  
## job splits as RLLLLLRLLLLL, improve= 4.131228, (0 missing)  
## campaign < 1.5 to the right, improve= 3.992527, (0 missing)  
## balance < 3316.5 to the left, improve= 3.380117, (0 missing)  
## duration < 538 to the right, improve= 3.088450, (0 missing)  
## Surrogate splits:  
## campaign < 2.5 to the right, agree=0.737, adj=0.464, (0 split)  
## duration < 285 to the right, agree=0.667, adj=0.321, (0 split)  
## job splits as RRRLRRRLRLRL, agree=0.649, adj=0.286, (0 split)  
## day < 29.5 to the left, agree=0.649, adj=0.286, (0 split)  
## balance < 192.5 to the left, agree=0.614, adj=0.214, (0 split)  
##   
## Node number 56: 76 observations  
## predicted class=no expected loss=0.3684211 P(node) =0.02401264  
## class counts: 48 28  
## probabilities: 0.632 0.368   
##   
## Node number 57: 10 observations  
## predicted class=yes expected loss=0.2 P(node) =0.003159558  
## class counts: 2 8  
## probabilities: 0.200 0.800   
##   
## Node number 78: 28 observations  
## predicted class=no expected loss=0.1785714 P(node) =0.008846761  
## class counts: 23 5  
## probabilities: 0.821 0.179   
##   
## Node number 79: 29 observations  
## predicted class=yes expected loss=0.2068966 P(node) =0.009162717  
## class counts: 6 23  
## probabilities: 0.207 0.793

**rpart.plot**(bankmodel1,main="Decision Tree for bank marketing data")



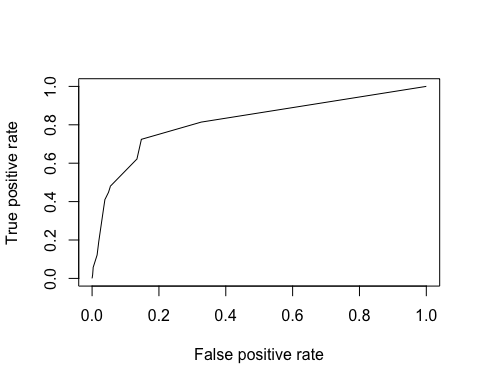
*# Making Prediction with decision tree*   
predictions<-**predict**(bankmodel1,testing,type = "class")  
**confusionMatrix**(predictions,testing$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1158 98  
## yes 42 58  
##   
## Accuracy : 0.8968   
## 95% CI : (0.8793, 0.9124)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.09211   
##   
## Kappa : 0.3991   
## Mcnemar's Test P-Value : 3.346e-06   
##   
## Sensitivity : 0.9650   
## Specificity : 0.3718   
## Pos Pred Value : 0.9220   
## Neg Pred Value : 0.5800   
## Prevalence : 0.8850   
## Detection Rate : 0.8540   
## Detection Prevalence : 0.9263   
## Balanced Accuracy : 0.6684   
##   
## 'Positive' Class : no   
##

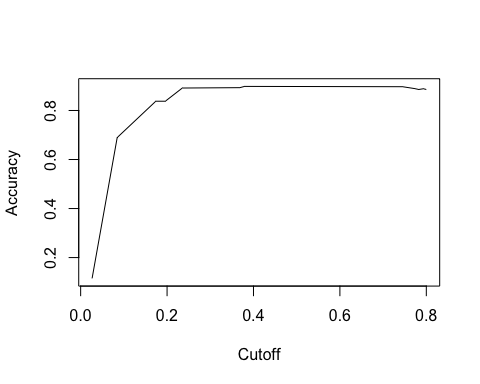
**varImp**(bankmodel1)

## Overall  
## age 5.923466  
## balance 12.231944  
## campaign 5.833314  
## contact 3.336733  
## day 11.643294  
## duration 154.771557  
## job 35.983792  
## loan 3.336733  
## marital 10.835453  
## month 97.774006  
## pdays 60.005245  
## poutcome 111.365011  
## previous 48.228430  
## education 0.000000  
## default 0.000000  
## housing 0.000000

*#Accuracy 89.75%*  
  
*#ROC Curve*  
pred.tree<-**predict**(bankmodel1,newdata = testing,type = "prob")[,2]  
pred2<-**prediction**(pred.tree,testing$y)  
pred2<-**prediction**(pred.tree,testing$y)  
**plot**(**performance**(pred2,"tpr","fpr"))



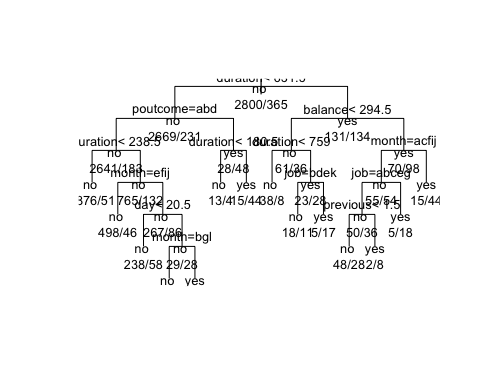
**plot**(**performance**(pred2,"acc"))



*#prunning the tree*  
**set.seed**(3)  
  
*#Selection right cp*  
bankmodel1$cptable[**which.min**(bankmodel1$cptable[,"xerror"]),"CP"]

## [1] 0.01

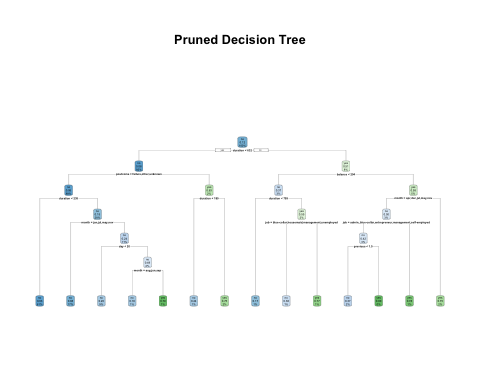
bankmodelprune<-**prune**(bankmodel1,cp=0.01)  
*#plotting the pruned tree.*   
**plot**(bankmodelprune,uniform = TRUE)  
**text**(bankmodelprune, use.n=TRUE, all=TRUE, cex=.8)



*#Prediction with prunned model*  
predictprune<-**predict**(bankmodelprune,testing,type = "class")  
**confusionMatrix**(predictprune,testing$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1158 98  
## yes 42 58  
##   
## Accuracy : 0.8968   
## 95% CI : (0.8793, 0.9124)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.09211   
##   
## Kappa : 0.3991   
## Mcnemar's Test P-Value : 3.346e-06   
##   
## Sensitivity : 0.9650   
## Specificity : 0.3718   
## Pos Pred Value : 0.9220   
## Neg Pred Value : 0.5800   
## Prevalence : 0.8850   
## Detection Rate : 0.8540   
## Detection Prevalence : 0.9263   
## Balanced Accuracy : 0.6684   
##   
## 'Positive' Class : no   
##

**rpart.plot**(bankmodelprune,main="Pruned Decision Tree")



**varImp**(bankmodelprune)

## Overall  
## age 5.923466  
## balance 12.231944  
## campaign 5.833314  
## contact 3.336733  
## day 11.643294  
## duration 154.771557  
## job 35.983792  
## loan 3.336733  
## marital 10.835453  
## month 97.774006  
## pdays 60.005245  
## poutcome 111.365011  
## previous 48.228430  
## education 0.000000  
## default 0.000000  
## housing 0.000000

*#Same accuracy*

**Model3: Random Forest**

Next we tried random forest and got the best prediction accuracy. We also plot the most important variables that were used to create the random forest.

*#RANDOM FOREST - Highest Accuracy*  
**library**(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':  
##   
## outlier

## The following object is masked from 'package:ggplot2':  
##   
## margin

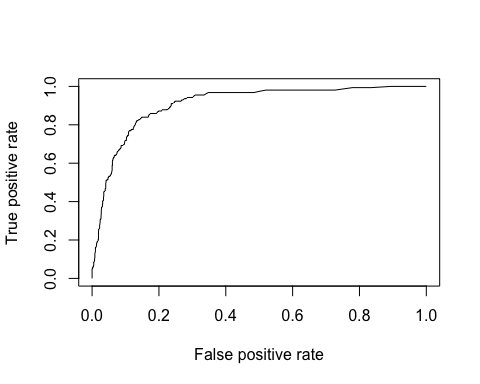
bankmodelrandom<-**randomForest**(y~.,data=training)  
**print**(bankmodelrandom)

##   
## Call:  
## randomForest(formula = y ~ ., data = training)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 9.83%  
## Confusion matrix:  
## no yes class.error  
## no 2723 77 0.0275000  
## yes 234 131 0.6410959

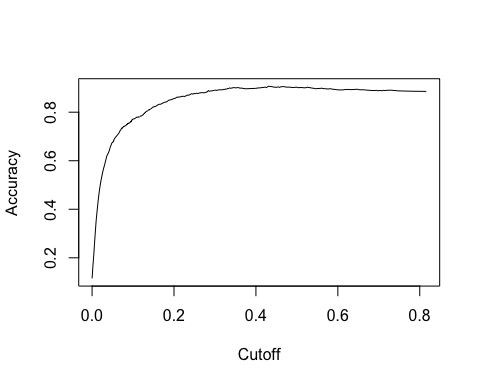
*#testing random forest*  
predictrandom<-**predict**(bankmodelrandom,testing,type = "class")  
**confusionMatrix**(predictrandom,testing$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1162 96  
## yes 38 60  
##   
## Accuracy : 0.9012   
## 95% CI : (0.884, 0.9165)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.03162   
##   
## Kappa : 0.421   
## Mcnemar's Test P-Value : 8.477e-07   
##   
## Sensitivity : 0.9683   
## Specificity : 0.3846   
## Pos Pred Value : 0.9237   
## Neg Pred Value : 0.6122   
## Prevalence : 0.8850   
## Detection Rate : 0.8569   
## Detection Prevalence : 0.9277   
## Balanced Accuracy : 0.6765   
##   
## 'Positive' Class : no   
##

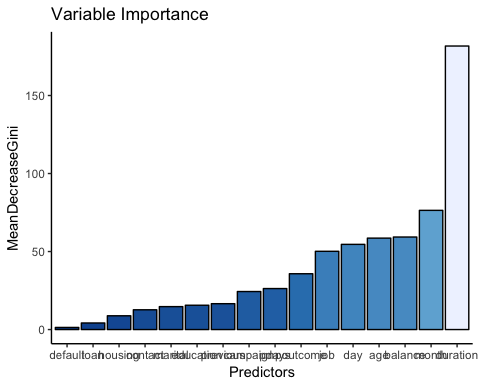
*#important variables*  
x<-**importance**(bankmodelrandom)  
  
*#ROC*  
pred.random<-**predict**(bankmodelrandom,newdata = testing,type = "prob")[,2]  
pred3<-**prediction**(pred.random,testing$y)  
**plot**(**performance**(pred3,"tpr","fpr"))



**plot**(**performance**(pred3,"acc"))



MeanDecreaseGini<-**c**(58.593336,50.150444,14.696016,15.584567,1.353045,59.300155,8.820523,4.206255,12.665181,54.595330,76.391365,181.712629,24.368490,26.272731,16.575289,35.764800)  
predictor<-**c**("age","job","marital","education","default","balance","housing","loan","contact","day","month","duration","campaign","pdays","previous","poutcome")  
x<-**data.frame**(predictor,MeanDecreaseGini)  
g<-**ggplot**(x,**aes**(**reorder**(x$predictor,x$MeanDecreaseGini),x$MeanDecreaseGini))  
plot1<-g+**geom\_col**(**aes**(fill=x$MeanDecreaseGini),colour="black")+**scale\_fill\_distiller**(palette = "Blues")+**labs**(title="Variable Importance",x="Predictors",y="MeanDecreaseGini")+**theme\_classic**()+**guides**(fill="none")  
plot1



**Model optimization**

**Improving accuracy with Bagging and boosting**

Accuracy actually went down, but we plot the variable importance for both the models.

##Bagging  
bankmodelrfbag<-**randomForest**(y~.,data = training,mtry=16,importance = TRUE)  
predictbag<-**predict**(bankmodelrfbag,testing,type = "class")  
**confusionMatrix**(predictbag,testing$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 1145 86  
## yes 55 70  
##   
## Accuracy : 0.896   
## 95% CI : (0.8785, 0.9118)  
## No Information Rate : 0.885   
## P-Value [Acc > NIR] : 0.10744   
##   
## Kappa : 0.441   
## Mcnemar's Test P-Value : 0.01152   
##   
## Sensitivity : 0.9542   
## Specificity : 0.4487   
## Pos Pred Value : 0.9301   
## Neg Pred Value : 0.5600   
## Prevalence : 0.8850   
## Detection Rate : 0.8444   
## Detection Prevalence : 0.9078   
## Balanced Accuracy : 0.7014   
##   
## 'Positive' Class : no   
##

*#Accuracy down*  
**varImp**(bankmodelrfbag)

## no yes  
## age 8.3434521 8.3434521  
## job 3.0548303 3.0548303  
## marital 9.7403698 9.7403698  
## education 3.2043821 3.2043821  
## default 11.4148454 11.4148454  
## balance 1.3657628 1.3657628  
## housing 7.8330301 7.8330301  
## loan -0.3202019 -0.3202019  
## contact 0.2137147 0.2137147  
## day 21.2437614 21.2437614  
## month 33.9860600 33.9860600  
## duration 73.4486810 73.4486810  
## campaign 0.6199988 0.6199988  
## pdays 17.1410625 17.1410625  
## previous 9.7038749 9.7038749  
## poutcome 32.4352603 32.4352603

*#Boosting*  
**library**(gbm)

## Warning: package 'gbm' was built under R version 3.3.2

## Loading required package: survival

##   
## Attaching package: 'survival'

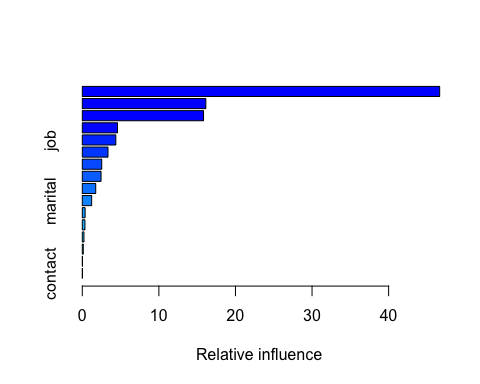
## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3

bankmodelboosted<-**gbm**(y~.,data=training,distribution = "multinomial",interaction.depth = 4,n.trees = 3000)  
**summary**(bankmodelboosted)



## var rel.inf  
## duration duration 46.6668109  
## month month 16.1321179  
## poutcome poutcome 15.8333809  
## pdays pdays 4.5932711  
## job job 4.3792518  
## day day 3.3566217  
## balance balance 2.5386102  
## age age 2.4497247  
## previous previous 1.7539019  
## marital marital 1.2222261  
## housing housing 0.3675322  
## education education 0.3442003  
## campaign campaign 0.2178374  
## loan loan 0.1281135  
## default default 0.0163995  
## contact contact 0.0000000

**Appendix Clustering Rcode**

set.seed(1680) # for reproducibility

library(dplyr) # for data cleaning

library(ISLR) # for college dataset

library(cluster) # for gower similarity and pam

library(Rtsne) # for t-SNE plot

library(ggplot2) # for visualization

library(magrittr)

library(cluster)

###############

#use the dat tranformed int eh proj1tranform file

no <- which(dat$y=="no")

yes <- which(dat$y=="yes")

new <- dat[c(sample(no,521,replace = F),yes),]

new <- subset(new, select = c(age,job,housing,month,duration,prev\_cat,poutcome,y,default))

#now get the gower\_dist distance

gower\_dist <- daisy(new,

metric = "gower",

)

######now choose optimal number of clusters

sil\_width <- c(NA)

for(i in 2:10){

pam\_fit <- pam(gower\_dist,

diss = TRUE,

k = i)

sil\_width[i] <- pam\_fit$silinfo$avg.width

}

plot(1:10, sil\_width,

xlab = "Number of clusters",

ylab = "Silhouette Width")

lines(1:10, sil\_width)

pam\_fit <- pam(gower\_dist, diss = TRUE, k = 2)

############# get summary

pam\_fit <- pam(gower\_dist, diss = TRUE, k = 2)

pam\_results <-new %>%

mutate(cluster = pam\_fit$clustering) %>%

group\_by(cluster) %>%

do(the\_summary = summary(.))

pam\_results$the\_summary

######now its visualization

tsne\_obj <- Rtsne(gower\_dist, is\_distance = TRUE)

tsne\_data <- tsne\_obj$Y %>%

data.frame() %>%

setNames(c("X", "Y")) %>%

mutate(cluster = factor(pam\_fit$clustering))

ggplot(aes(x = X, y = Y), data = tsne\_data) +

geom\_point(aes(color = cluster))

##########now I want to get the cluster for the customer's background info

#dat[,1:8] columns for all the background info

new <- subset(dat,select = c(age,job,housing,month,contact,duration,y,poutcome,prev\_cat,education,marital))

new <- subset(dat, select = c(duration,month,bal\_cat,age,job,poutcome,y))

new <- subset(dat, select = c(age,job,default,prev\_cat,housing,month,duration,poutcome,y))

gower\_dist <- daisy(new,

metric = "gower"

)

######now choose optimal number of clusters

sil\_width <- c(NA)

for(i in 2:10){

pam\_fit <- pam(gower\_dist,

diss = TRUE,

k = i)

sil\_width[i] <- pam\_fit$silinfo$avg.width

}

plot(1:10, sil\_width,

xlab = "Number of clusters",

ylab = "Silhouette Width")

lines(1:10, sil\_width)

############# get summary

pam\_fit <- pam(gower\_dist, diss = TRUE, k = 3)

pam\_results <-new %>%

mutate(cluster = pam\_fit$clustering) %>%

group\_by(cluster) %>%

do(the\_summary = summary(.))

pam\_results$the\_summary

#VISUALIZE

tsne\_obj <- Rtsne(gower\_dist, is\_distance = TRUE)

tsne\_data <- tsne\_obj$Y %>%

data.frame() %>%

setNames(c("X", "Y")) %>%

mutate(cluster = factor(pam\_fit$clustering))

ggplot(aes(x = X, y = Y), data = tsne\_data) +

geom\_point(aes(color = cluster))

**Appendix Data transforming**

dat <- read.csv("bank.csv")

##########first take log of the age column

dat$age <- log(dat$age)

dat$duration <- log(dat$duration)

no <- dat[which(dat$y=='no'),]

yes <- dat[which(dat$y=='yes'),]

ggplot(no,aes(x = job)) + geom\_bar()

ggplot(yes,aes(x = job)) + geom\_bar()

######dealing with the previous column, divide it into three categories

B <- which(dat$previous>3)

B <- which(dat$previous<3 & dat$previous>0)

length(B)

#divide equal to number of calling = 0, class: never called before, 3705

#1,2, class: called before 479

#>3, class: called multiple times.224

dat$prev\_cat <- dat$previous

dat$prev\_cat[which(dat$previous>=3)] = "Multiple"

dat$prev\_cat[which(dat$previous==0)] = "never"

dat$prev\_cat[which(dat$previous<3 & dat$previous>0)] = "called"

dat$prev\_cat <- as.factor(dat$prev\_cat)

#dat <- dat[,-15]

#########now start with the campign column

ggplot(dat,aes(x = campaign)) + geom\_bar()

table(dat$campaign)

dat$camp\_cat <- dat$campaign

dat$camp\_cat[which(dat$campaign>=3)]= "Multiple"

dat$camp\_cat[which(dat$campaign<3)]= "called"

#dat <- dat[,-13]

dat$camp\_cat <- as.factor(dat$camp\_cat)

############now with the balance

hist(dat$balance)

length(which(dat$balance>0 & dat$balance < 5000))

dat$bal\_cat <- dat$balance

dat$bal\_cat[which(dat$balance>=0 & dat$balance <= 5000)] = "has balance"

dat$bal\_cat[which(dat$balance<0)] = "negative balance"

dat$bal\_cat[which(dat$balance>5000)] = "high balance"

dat$bal\_cat <- as.factor(dat$bal\_cat)

#############pdays

ggplot(dat,aes(x = pdays)) + geom\_histogram()

hist(dat$pdays[dat$pdays!=-1])

dat$pday\_cat <- dat$pdays

dat$pday\_cat[which(dat$pdays==-1)] = "no before"

dat$pday\_cat[which(dat$pdays>0 & dat$pdays<200)] = "200-400days"

dat$pday\_cat[which(dat$pdays>200)] = "after 200days"

dat$pday\_cat <- as.factor(dat$pday\_cat)

#now delete the original balance, pdays, day, preivous,campign

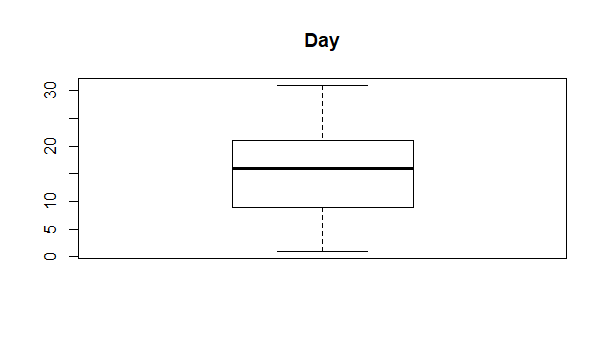
#try delete the day column first

dat <- subset(dat,select = -c(pdays,previous,campaign,balance))

**Appendix Boxplots**

## bmdbalance<-boxplot(bmd$balance, main = "Balance")

## bmdday<-boxplot(bmd$day, main = "Day")



bmdcampaign<-boxplot(bmd$campaign, main = "Campaign")

