Table of Contents

Introduction:

Whether it's a banking industry or an online shopping platform all businesses want to increase their customer base. It is always important for telemarketing department to approach potential customers in a correct manner to convert them to real customers. There are different factors which converts the potential customers to real customers. The undertaken project is about to predict if the client of the bank will subscribe to a term deposit when approached by telemarketing group. This project will help businesses to understand the different factors to improve their conventional telemarketing approach and increase the efficiency of marketing group. So, different classification models are built using R programming language and comparison has been done, to address above written issues.

Data Set Abstract:

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y). The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The data set has 4521 observations and 17 variables including the dependent variable, which explains has the client subscribed a term deposit (binary: 'yes' or 'no'). Variables of our analysis and their description:

Independent variables:

1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no', 'yes', 'unknown')
6 - housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
7 - loan: has personal loan? (categorical: 'no' 'yes' 'unknown')
# related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular', 'telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue', 'wed', 'thu', 'fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not

known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure', 'nonexistent', 'success')

16 – balance: How much money the customer has in his bank account

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

Below we can observe how the actual data set looks like. The categorical variables are converted to factors, to realize the labels they contain.

```
## 'data.frame':    4521 obs. of  17 variables:
##  $ age      : int  30 33 35 30 59 35 36 39 41 43 ...
##  $ job      : Factor w/ 12 levels "admin.","blue-collar",..: 11 8 5 5 2 5 7 10 3 8
##  $ marital  : Factor w/ 3 levels "divorced","married",..: 2 2 3 2 2 3 2 2 2 2 ...
##  $ education: Factor w/ 4 levels "primary","secondary",..: 1 2 3 3 2 3 3 2 3 1 ...
##  $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : int  1787 4789 1350 1476 0 747 307 147 221 -88 ...
##  $ housing  : Factor w/ 2 levels "no","yes": 1 2 2 2 2 1 2 2 2 2 ...
##  $ loan     : Factor w/ 2 levels "no","yes": 1 2 1 2 1 1 1 1 1 2 ...
##  $ contact  : Factor w/ 3 levels "cellular","telephone",..: 1 1 1 3 3 1 1 1 3 1 ..
##  $ day      : int  19 11 16 3 5 23 14 6 14 17 ...
##  $ month    : Factor w/ 12 levels "apr","aug","dec",..: 11 9 1 7 9 4 9 9 9 1 ...
##  $ duration : int  79 220 185 199 226 141 341 151 57 313 ...
##  $ campaign : int  1 1 1 4 1 2 1 2 2 1 ...
##  $ pdays    : int  -1 339 330 -1 -1 176 330 -1 -1 147 ...
##  $ previous : int  0 4 1 0 0 3 2 0 0 2 ...
##  $ poutcome : Factor w/ 4 levels "failure","other",..: 4 1 1 4 4 1 2 4 4 1 ...
##  $ y        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

Data Manipulation and Exploratory Analysis:

To understand the distribution of the data and to know the count of different classes of the output or dependent variable, the proportion of “yes” and “no” has computed. 88.47% of the dependent variable has a class of “no” and 11.52% of the dependent variable has a class of “yes”. This indicates that data is imbalanced.

```
##      no     yes
## 0.88476 0.11524
```

Further it is checked if there is any NA value presented in complete data set and in any independent variable column. It is found that there is no NA in any independent variable column, hence in the whole data set.

```
## [1] 0

##  age job marital education default balance housing loan contact
##   0   0    0         0       0       0       0           0      0
##  day     month  duration  campaign  pdays previous  poutcome    y
##   0        0         0        0       0       0         0       0
```

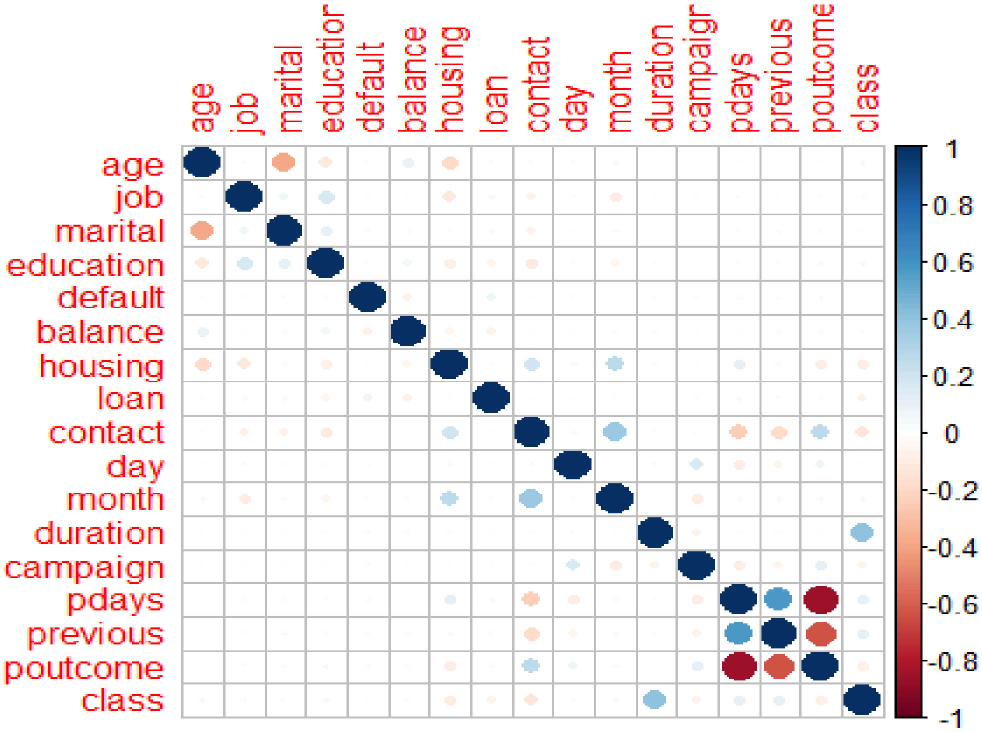
To analyze the data set and apply different concepts, categorical values have been converted to numbers, where each number signifies a class of the categorical variable, and numeric variables have been scaled or standardized to reduce the effect of their unit of measurement. Below we can see the final modified or manipulated data set.

```
## 'data.frame':    4521 obs. of  16 variables:
##  $ age      : num [1:4521, 1] -1.056 -0.772 -0.583 -1.056 1.686 ...
##   ..- attr(*, "scaled:center")= num 41.2
##   ..- attr(*, "scaled:scale")= num 10.6
##  $ job      : num  11 8 5 5 2 5 7 10 3 8 ...
##  $ marital  : num  2 2 3 2 2 3 2 2 2 2 ...
##  $ education: num  1 2 3 3 2 3 3 2 3 1 ...
##  $ default  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : num [1:4521, 1] 0.1211 1.1185 -0.0241 0.0177 -0.4727 ...
##   ..- attr(*, "scaled:center")= num 1423
##   ..- attr(*, "scaled:scale")= num 3010
##  $ housing  : num  1 2 2 2 2 1 2 2 2 2 ...
##  $ loan     : num  1 2 1 2 1 1 1 1 1 2 ...
##  $ contact  : num  1 1 1 3 3 1 1 1 3 1 ...
##  $ day      : int  19 11 16 3 5 23 14 6 14 17 ...
##  $ month    : num  11 9 1 7 9 4 9 9 9 1 ...
##  $ duration : num [1:4521, 1] -0.712 -0.169 -0.304 -0.25 -0.146 ...
##   ..- attr(*, "scaled:center")= num 264
##   ..- attr(*, "scaled:scale")= num 260
##  $ campaign : num [1:4521, 1] -0.577 -0.577 -0.577 0.388 -0.577 ...
##   ..- attr(*, "scaled:center")= num 2.79
##   ..- attr(*, "scaled:scale")= num 3.11
##  $ pdays    : num [1:4521, 1] -0.407 2.989 2.899 -0.407 -0.407 ...
##   ..- attr(*, "scaled:center")= num 39.8
##   ..- attr(*, "scaled:scale")= num 100
##  $ previous : num [1:4521, 1] -0.32 2.04 0.27 -0.32 -0.32 ...
##   ..- attr(*, "scaled:center")= num 0.543
##   ..- attr(*, "scaled:scale")= num 1.69
##  $ poutcome : num  4 1 1 4 4 1 2 4 4 1 ...
```

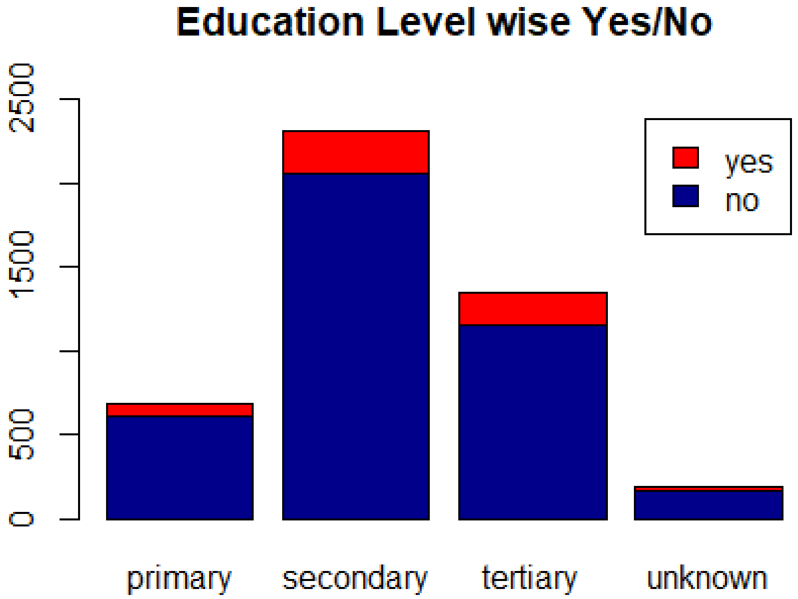
Before proceeding to any analysis, the correlation between all the variables have been computed, which can be observed below. From the correlation matrix we can observe that most of the variables are very weakly correlated. There is some correlation between poutcome pdays & previous, and between pdays & previous. (Dependent variable is class.)

```
##             age   job marital education default balance housing  loan
## age        1.00 -0.02   -0.38     -0.12   -0.02    0.08   -0.19 -0.01
## job       -0.02  1.00    0.07      0.17    0.01    0.01   -0.13 -0.04
## marital   -0.38  0.07    1.00      0.10   -0.02    0.02   -0.03 -0.05
## education -0.12  0.17    0.10      1.00   -0.01    0.06   -0.09 -0.05
## default   -0.02  0.01   -0.02     -0.01    1.00   -0.07    0.01  0.06
## balance    0.08  0.01    0.02      0.06   -0.07    1.00   -0.05 -0.07
## housing   -0.19 -0.13   -0.03     -0.09    0.01   -0.05    1.00  0.02
## loan      -0.01 -0.04   -0.05     -0.05    0.06   -0.07    0.02  1.00
## contact    0.02 -0.07   -0.07     -0.11    0.01   -0.01    0.20 -0.01
## day       -0.02  0.01    0.01      0.01   -0.01   -0.01   -0.03  0.00
## month     -0.04 -0.10   -0.04     -0.05    0.01    0.02    0.27  0.02
## duration   0.00 -0.01    0.01     -0.01   -0.01   -0.02    0.02  0.00
## campaign  -0.01  0.00    0.01      0.00   -0.01   -0.01    0.00  0.02
## pdays     -0.01 -0.02    0.02      0.01   -0.03    0.01    0.12 -0.03
## previous   0.00  0.01    0.04      0.02   -0.03    0.03    0.04 -0.02
## poutcome  -0.01  0.01   -0.03     -0.03    0.04   -0.03   -0.09  0.03
## class      0.05  0.03    0.02      0.04    0.00    0.02   -0.10 -0.07
##           contact   day month duration campaign pdays previous poutcome  class
## age          0.02 -0.02 -0.04     0.00    -0.01 -0.01     0.00    -0.01   0.05
## job         -0.07  0.01 -0.10    -0.01     0.00 -0.02     0.01     0.01   0.03
## marital     -0.07  0.01 -0.04     0.01     0.01  0.02     0.04    -0.03   0.02
## education   -0.11  0.01 -0.05    -0.01     0.00  0.01     0.02    -0.03   0.04
## default      0.01 -0.01  0.01    -0.01    -0.01 -0.03    -0.03     0.04   0.00
## balance     -0.01 -0.01  0.02    -0.02    -0.01  0.01     0.03    -0.03   0.02
## housing      0.20 -0.03  0.27     0.02     0.00  0.12     0.04    -0.09  -0.10
## loan        -0.01  0.00  0.02     0.00     0.02 -0.03    -0.02     0.03  -0.07
## contact      1.00 -0.03  0.37    -0.01     0.01 -0.24    -0.19     0.27  -0.13
## day         -0.03  1.00 -0.01    -0.02     0.16 -0.09    -0.06     0.07  -0.01
## month        0.37 -0.01  1.00     0.00    -0.11  0.03     0.05    -0.03  -0.04
## duration    -0.01 -0.02  0.00     1.00    -0.07  0.01     0.02     0.00   0.40
## campaign     0.01  0.16 -0.11    -0.07     1.00 -0.09    -0.07     0.11  -0.06
## pdays       -0.24 -0.09  0.03     0.01    -0.09  1.00     0.58    -0.86   0.10
## previous    -0.19 -0.06  0.05     0.02    -0.07  0.58     1.00    -0.64   0.12
## poutcome     0.27  0.07 -0.03     0.00     0.11 -0.86    -0.64     1.00  -0.08
## class       -0.13 -0.01 -0.04     0.40    -0.06  0.10     0.12    -0.08   1.00
```

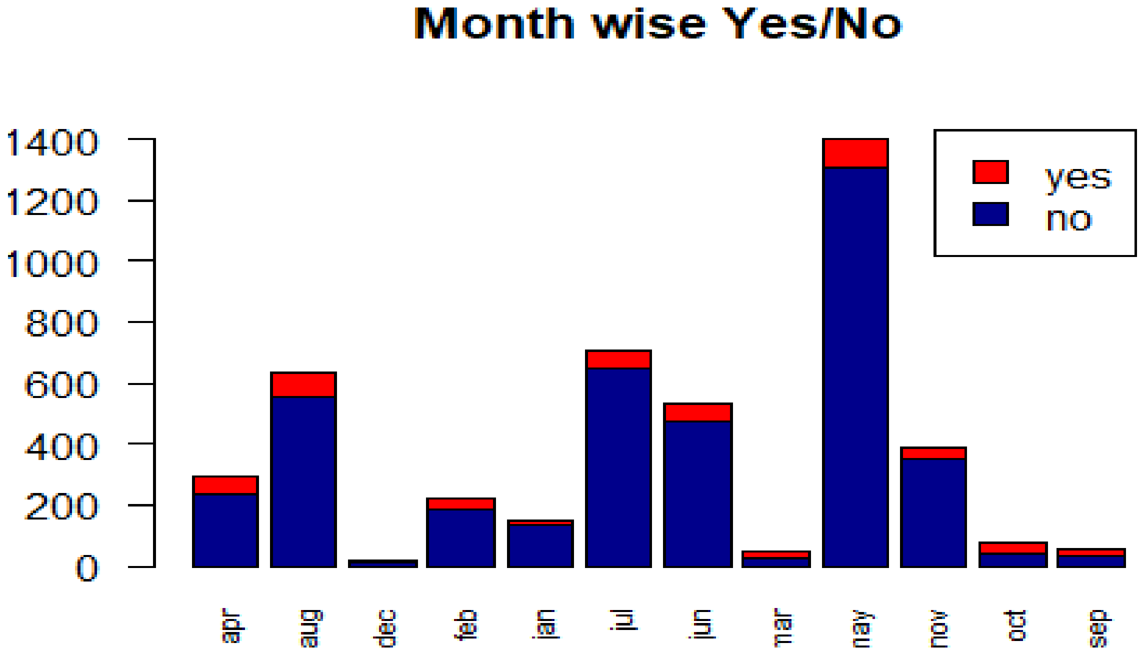
For better vizualization below correlation matrix is produced, where circle diameter represents the absolute correlation value and color scale represents sign of the correlation.



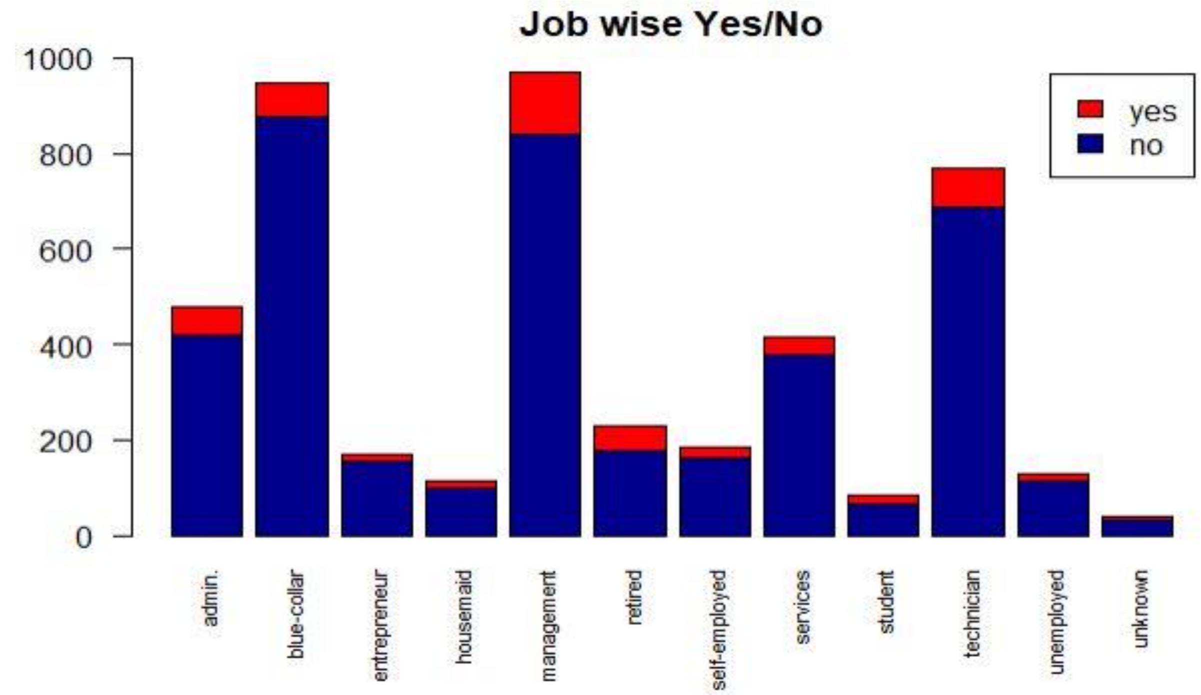
To understand what factors really matter for a customer of subscribe for a term deposit, some exploratory analysis has been conducted. Below graph says that the marketing team has approached the most of customers, who have a secondary level of eductaion and high proportion of subscription for the term deposit (5.4 %) in compare to the other customers of different education level.



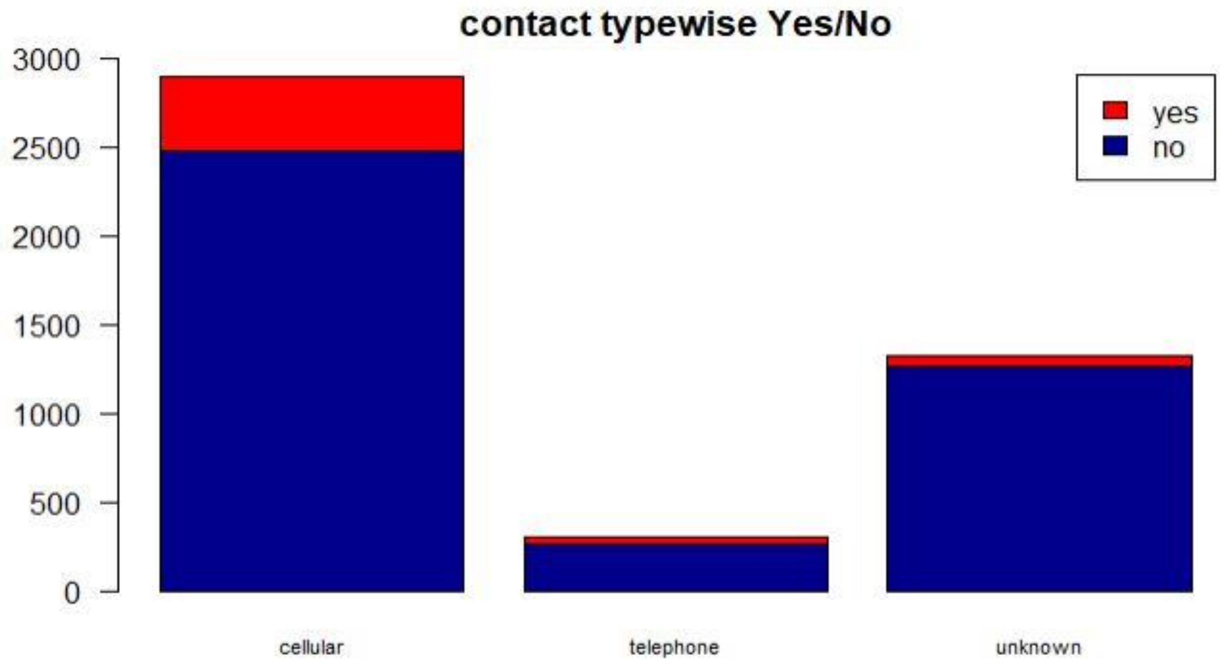
It also matters at what month the marketing team approached the customers. The below graph represents that most of the customers have been approached in the month of the May and there is a higher chance that they will subscribe for the term deposit.



Beginning of the financial year could be a reason for customers being in better financial conditions and interested in the term deposit.



Above graph explains that almost same number of customers having blue-collar and management job have been approached, but more number of customers having management job have been subscribed for the term deposit than number of customers having blue-collar job.



From the above graph it can be observed that most of the customers have been reached over cellular and at the same time proportion of customers subscribing for the term deposit is higher when reached over cellular. We can conduct more of this type of exploratory analysis to understand how independent variables are related to the dependent variables.

Data Analysis (Models for Classification and Comparison):

To apply different concepts and to predict whether customers approached will subscribe for the term deposit, data has been divided in training data set to develop the model and test data set for classification and accuracy check. For the accuracy check and comparison purpose, confusion matrix has been used for all the models. Whole data is partition is in a 70:30 ratio. Train data set has 70 % of the whole data set and contains 3164 observations. (Below given the structure of training data set)

```
## 'data.frame':    3164 obs. of  16 variables:
##  $ age      : num [1:3164, 1] 0.173 -0.962 -0.867 0.268 -1.245 ...
##  $ job      : num  5 8 8 2 8 3 2 6 8 2 ...
##  $ marital  : num  2 3 3 2 2 2 2 2 3 2 ...
##  $ education: num  3 2 2 1 2 2 2 1 1 2 ...
##  $ default  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : num [1:3164, 1] -0.472 -0.462 -0.457 -0.383 -0.473 ...
##  $ housing  : num  1 2 2 2 2 1 2 1 2 2 ...
##  $ loan     : num  2 2 1 2 1 1 2 1 1 1 ...
##  $ contact  : num  1 3 1 3 1 1 1 1 2 1 ...
##  $ day      : int  20 26 20 9 5 6 30 6 13 14 ...
##  $ month    : num  2 9 1 9 9 4 6 2 9 10 ...
##  $ duration : num [1:3164, 1] 0.801 0.912 -0.985 -0.654 0.273 ...
##  $ campaign : num [1:3164, 1] -0.2552 -0.2552 0.0664 -0.5768 -0.2552 ...
##  $ pdays    : num [1:3164, 1] -0.407 -0.407 -0.407 -0.407 2.409 ...
##  $ previous : num [1:3164, 1] -0.32 -0.32 -0.32 -0.32 4.4 ...
##  $ poutcome : num  4 4 4 4 1 4 4 4 4 4 ...
```

Test data set has 30 % of the whole data set and contains 1357 observations. (Below given the structure of test data set)

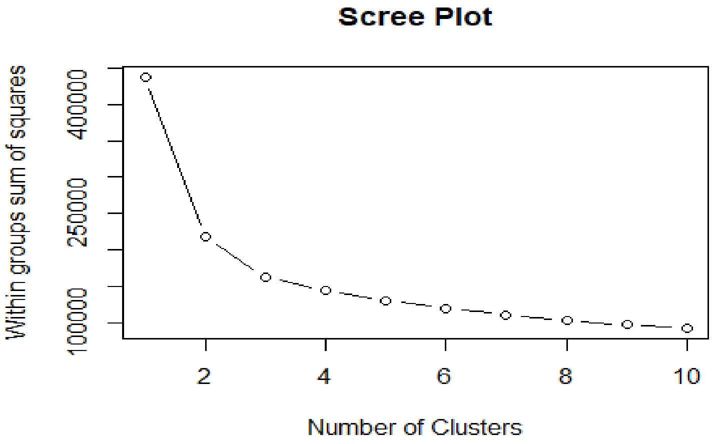
```
## 'data.frame':    1357 obs. of  16 variables:
##  $ age      : num [1:1357, 1] -1.056 -0.772 1.686 -0.205 -0.205 ...
##  $ job      : num  11 8 2 10 8 10 8 5 3 4 ...
##  $ marital  : num  2 2 2 2 2 2 2 1 2 2 ...
##  $ education: num  1 2 2 2 2 2 2 4 2 3 ...
##  $ default  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ balance  : num [1:1357, 1] 0.121 1.119 -0.473 -0.424 2.642 ...
##  $ housing  : num  1 2 2 2 2 1 1 2 1 1 ...
##  $ loan     : num  1 2 1 1 1 1 1 1 1 1 ...
##  $ contact  : num  1 1 3 1 3 1 1 1 1 1 ...
##  $ day      : int  19 11 5 6 20 27 7 18 7 30 ...
##  $ month    : num  11 9 9 9 9 2 6 10 6 5 ...
##  $ duration : num [1:1357, 1] -0.7118 -0.1692 -0.1461 -0.4347 0.0348 ...
##  $ campaign : num [1:1357, 1] -0.577 -0.577 -0.577 -0.255 -0.577 ...
##  $ pdays    : num [1:1357, 1] -0.407 2.989 -0.407 -0.407 -0.407 ...
##  $ previous : num [1:1357, 1] -0.32 2.04 -0.32 -0.32 -0.32 ...
##  $ poutcome : num  4 1 4 4 4 4 2 4 4 4 ...
```

As, shown previously, the data set is imbalanced according to the proportion of the classes ("yes" or "no") in the outcome variable. So, observations are randomly selected for train and test data set to maintain the same ratio and proportions are presented below.

```
## train.labels                           ## test.labels
##        no       yes                    ##        no       yes
## 0.8843236 0.1156764                    ## 0.8857775 0.1142225
```

Model 1: K (3)-NN:

The first model considered for classification is k- nearest neighbor. An elbow test conducted to find out the best K. From the below scree plot k=3 is the best one, as there is no big change in slope after this k. So, model is developed based on k=3.



Below given the summary statistics of the k (3)-NN model.

```
## Confusion Matrix and Statistics
##           Reference
## Prediction   no  yes
##        no  1161  149
##        yes   41    6
##
##                Accuracy : 0.86
##                  95% CI : (0.8404, 0.878)
##     No Information Rate : 0.8858
##     P-Value [Acc > NIR] : 0.9984
##
##                   Kappa : 0.0066
##  Mcnemar's Test P-Value : 8.321e-15
##
##             Sensitivity : 0.96589
##             Specificity : 0.03871
##          Pos Pred Value : 0.88626
##          Neg Pred Value : 0.12766
##              Prevalence : 0.88578
##          Detection Rate : 0.85556
##    Detection Prevalence : 0.96536
##       Balanced Accuracy : 0.50230
##        'Positive' Class : no
```

Model accuracy, which defines how often the classifier is correct, is 86%. When “no” is considered as the positive class, model Sensitivity is 96.59%, which signifies percentage of the “no” the model can identify correctly. Model Specificity is 3.87%, which signifies percentage of the “yes” the model can identify correctly. Also, K (4)-NN has been implemented, but there is no significant improvement in accuracy, sensitivity, and specificity.

Model 2(a): Logistic Regression (All variables considered)

Next model considered is logistic classification model. Below given is the model summary.

```
## Call:
## glm(formula = convert ~ ., family = binomial(logit), data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.7725  -0.4393  -0.3050  -0.1908   3.0316  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.015335   0.814780  -1.246 0.212711    
## age          0.145395   0.065629   2.215 0.026733 *  
## job          0.022317   0.020483   1.090 0.275925    
## marital      0.021019   0.116446   0.181 0.856758    
## education    0.075246   0.085537   0.880 0.379027    
## default      0.420087   0.497580   0.844 0.398524    
## balance      0.051720   0.051199   1.010 0.312407    
## housing     -0.791950   0.141702  -5.589 2.29e-08 ***
## loan        -0.865561   0.225306  -3.842 0.000122 ***
## contact     -0.509137   0.096487  -5.277 1.31e-07 ***
## day          0.003748   0.008086   0.464 0.642953    
## month        0.016433   0.021616   0.760 0.447113    
## duration     1.041673   0.059139  17.614  < 2e-16 ***
## campaign    -0.190885   0.091013  -2.097 0.035964 *  
## pdays        0.288370   0.091020   3.168 0.001534 ** 
## previous     0.158585   0.055962   2.834 0.004600 ** 
## poutcome     0.156181   0.107064   1.459 0.144631    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2263  on 3164  degrees of freedom
## Residual deviance: 1715  on 3148  degrees of freedom
## AIC: 1749
## 
## Number of Fisher Scoring iterations: 6
```

Below given is the model output summary statistics.

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   no  yes
##        no  1137   94
##        yes   63   62
## 
```

```
##                Accuracy : 0.8842
##                  95% CI : (0.866, 0.9008)
##     No Information Rate : 0.885
##     P-Value [Acc > NIR] : 0.55504
##
##                   Kappa : 0.3776
##  Mcnemar's Test P-Value : 0.01665
##
##             Sensitivity : 0.9475
##             Specificity : 0.3974
##          Pos Pred Value : 0.9236
##          Neg Pred Value : 0.4960
##              Prevalence : 0.8850
##          Detection Rate : 0.8385
##    Detection Prevalence : 0.9078
##       Balanced Accuracy : 0.6725
##        'Positive' Class : no
```

Model accuracy, which defines how often the classifier is correct, is 88.42%, with a cut-off value of 0.3, as sigmoid function is very sensitive. When "no" is considered as the positive class, model Sensitivity is 94.75%, which signifies percentage of the "no" the model can identify correctly. Model Specificity is 39.74%, which signifies percentage of the "yes" the model can identify correctly. Some of the independent variables are not significant and have very less weight in the model.

Model 2(b): Logistic Regression (Significant variables & high weight variables)

A new logistic model has been built using variable which are significant and carries high weight in the previous model. Variables are age, education, housing, loan, contact, duration, campaign, pdays, previous, poutcome, and month. Below given model output statistics:

```
## Confusion Matrix and Statistics
##           Reference
## Prediction   no  yes
##        no  1139   92
##        yes   61   64
##
##                Accuracy : 0.8872
##                  95% CI : (0.8691, 0.9035)
##     No Information Rate : 0.885
##     P-Value [Acc > NIR] : 0.41984
##
##                   Kappa : 0.3934
##  Mcnemar's Test P-Value : 0.01529
##
##             Sensitivity : 0.9492
##             Specificity : 0.4103
##          Pos Pred Value : 0.9253
##          Neg Pred Value : 0.5120
##              Prevalence : 0.8850
##          Detection Rate : 0.8400
##    Detection Prevalence : 0.9078
##       Balanced Accuracy : 0.6797
##        'Positive' Class : no
```

Model accuracy, which defines how often the classifier is correct, is 88.72%, with a cut-off value of 0.3, as sigmoid function is very sensitive. When “no” is considered as the positive class, model Sensitivity is 94.92%, which signifies percentage of the “no” the model can identify correctly. Model Specificity is 41.03%, which signifies percentage of the “yes” the model can identify correctly. Though this model has similar output as previous model, it has higher specificity. Also, a logistic classification model with only significant independent variables has been developed, but it did not perform better than the previous two models. (So, not included in the report)

Model 3: Decision Tree (c50)

Next Decision Tree model is considered for classification. Below given is the model summary/output statistics:

```
## Confusion Matrix and Statistics
##           Reference
## Prediction   no  yes
##        no  1158   93
##        yes   42   63
##
##                Accuracy : 0.9004
##                  95% CI : (0.8833, 0.9159)
##     No Information Rate : 0.885
##     P-Value [Acc > NIR] : 0.03847
##
##                   Kappa : 0.43
##  Mcnemar's Test P-Value : 1.683e-05
##
##             Sensitivity : 0.9650
##             Specificity : 0.4038
##          Pos Pred Value : 0.9257
##          Neg Pred Value : 0.6000
##              Prevalence : 0.8850
##          Detection Rate : 0.8540
##    Detection Prevalence : 0.9226
##       Balanced Accuracy : 0.6844
##        'Positive' Class : no
```

Model accuracy, which defines how often the classifier is correct, is 90.04%, with 11 trials. When “no” is considered as the positive class, model Sensitivity is 96.50%, which signifies percentage of the “no” the model can identify correctly. Model Specificity is 40.38%, which signifies percentage of the “yes” the model can identify correctly. Also, a decision model has been developed using rpart package, but it did not perform better than this model. (So, not included in the report)

Model 4: Neural Network (neuralnet)

Next neural network model is considered for classification, using neural net package. The model has been developed with 3 hidden layers and a learning rate of 0.00001. Below given is the output summary statistics of the model:

```
## Confusion Matrix and Statistics
##           Reference
## Prediction    1    2
##          1 1143  118
##          2   42   54
##
##                 Accuracy : 0.8820929
##                   95% CI : (0.8637332, 0.8987785)
##      No Information Rate : 0.8732498
##      P-Value [Acc > NIR] : 0.174317
##
##                    Kappa : 0.343358
##   Mcnemar's Test P-Value : 0.000000003042833
##
##              Sensitivity : 0.9645570
##              Specificity : 0.3139535
##           Pos Pred Value : 0.9064235
##           Neg Pred Value : 0.5625000
##               Prevalence : 0.8732498
##           Detection Rate : 0.8422992
##     Detection Prevalence : 0.9292557
##        Balanced Accuracy : 0.6392552
##         'Positive' Class : 1
```

Model accuracy, which defines how often the classifier is correct, is 88.21 %. When “no” is considered as the positive class, model Sensitivity is 96.46%, which signifies percentage of the “no” the model can identify correctly. Model Specificity is 31.39%, which signifies percentage of the “yes” the model can identify correctly.

Apart from these models, some other models are also developed using machine learning concepts. Some of them are presented in this report and are presented in ***BLUE*** in the below given performance report.

Table: Performance Report of Various Models

Model	Accuracy	Sensitivity	Specificity
K (3)-NN	0.8651	0.9659	0.0387
LR - (All Variables)	0.8842	0.9475	0.3974
LR - (Cor & weight)	0.8872	0.9492	0.4103
LR – (Significant Variables)	0.8857	0.9508	0.3846
LDA	0.8850	0.9633	0.2821
SVM (ksvm)	0.8901	0.9908	0.1154
DT (c50)	0.9004	0.9650	0.4038
DT (rpart)	0.8938	0.9725	0.2885
Neural Net	0.8821	0.9646	0.3140

*R-Code for all the models has been provided in the appendix (1)

Models' Analysis:

In this undertaken project, it is very important to know the potential customers who will subscribe the term deposit and what factors will lead them to do so, rather than who will not. Because here our primary goal is to predict who will subscribe the term deposit and what are the attributes of the customer. As it can be observed from the above performance table, Sensitivity signifies percentage of the "no" the model can identify correctly, and Specificity signifies percentage of the "yes" the model can identify correctly, when positive class is "no". So, models with high accuracy and high Specificity are considered. K (3)-NN, LDA, SVM (ksvm), Neural Net, and Decision Tree(rpart) model have very low Specificity, so those are not considered for classification.

Logistic Regression model built with variable having higher correlation and higher weight has outperformed other logistic regression models. It has an accuracy of 88.72% and Specificity of 41.03%, with a cut-off value of 0.3. Observing the weight assigned to independent variables, it can be observed that a higher negative weight is assigned to campaign. That means more number of outbound calls have a negative effect on customers to subscribe for the term deposit. Duration has a high positive weight in the model, which means if the customers talks to representative for a longer time, the customer is more likely to subscribe the term deposit. Housing and loan has high negative weight, which means if the customer has a housing loan or personal loan, he or she will not subscribe the term deposit. Similarly, how the other attributes affect the customer decision for subscribing the term deposit can be interpreted.

Decision tree built using c50 package has outperformed all the models with an accuracy of 90.04% and a specificity of 40.38%. Details of tree branching, nodes, and leaf can be found in appendix (2). For this classification problem as per the analysis this model is recommended. This model has the benefit of high Sensitivity, while having a competitive Specificity when compared to the other models. C5.0 fits classification tree models or rule-based models using Quinlan's C5.0 algorithm. Internally, the code attempts to halt boosting if it appears to be ineffective. For this reason, the value of trials may be different from what the model produced. In this case for model building a trial of 11 has been used. With this model predicting which customer is not going to subscribe the term deposit is very accurate. So, knowing who is not going to subscribe is useful in one way for optimizing the resources. At the same time knowing who is going to subscribe the term deposit with an accuracy of 40.38% is useful to approach the potential customers only.

Future Analysis:

For future analysis, customers can be divided in different clusters using cluster analysis, could be as per their job, education level, or age range and then models can be built cluster wise for better classification or prediction. Also, different models could be built using other machine learning techniques like Random Forest with bagging and boosting, etc. to check their prediction capability. More attributes regarding the customers can be added to the analysis to understand various factors lead to making decision to subscribe the term deposit.

Conclusion:

Predicting marketing output in the banking industry is one of the most challenging issues because of the size of the transaction data. Bank marketing and decisions are critical for maintaining the relationship with the potential customers. To sustain in the competitive business environment, it is very important to understand customers and approach them in proper manner. Data mining, predictive analytics and text mining are the way for such marketing strategies. Above models developed using machine learning concepts can be used to increase the effectiveness of the campaign. Both the Logistic regression and Decision Tree models worked efficiently to classify the customers' decision to subscribe the term deposit. From the logistic regression model's independent variables' weights, we can understand the factors that influence the customers' decision for subscribing the term deposit. Decision tree model is good enough in predicting which customer will not subscribe for the term deposit and which customer will do, so overall accuracy is best of all the models. At this stage Decision tree developed using c50 will be considered to identify the customers decision for subscribing the term deposit.

References

- Moro, S., & Rita, P. (n.d.). UCI Machine Learning Repository: Bank Marketing Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/bank marketing
- S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014
- Shmueli, G., Bruce, P., & Patel, N. R. (n.d.). *Data Mining for Business Analytics: Concepts, Techniques, and Applications with XLMiner*(3rd ed.). Wiley.
- Groenfeldt, T. (2016, April 03). Bank Marketing: Dump Data Scientists -- Try Marketing By Walking Around. Retrieved from https://www.forbes.com/sites/tomgroenfeldt/2016/03/09/bank-marketing-dump-data-scientists-try-marketing-by-walking-around/#691ca4457cb9

Appendix

1.Code for the Data Modeling and Analysis

```r
#importing data from the system
a<-read.csv ("C:/Users/Rupam Das/Desktop/Project/bank.csv", sep = ";")
str(a)

#probability table of dependent variable
prop.table(table(a$y))

#checking number of nas
#checking NA values columnwise
sum(is.na(a))

sapply(a, function(x) sum(is.na(x)))# number of nas

# data manipulation and creating data sets for different analysis
#scaling the numerical variables
a1<-a[,-17]

for(i in c("age","balance","duration","campaign", "pdays", "previous")){
  a1[,i]<- scale(a1[,i])
}
#str(a1)

#converting the factors to numeric values
for(i in c("job","education","marital", "default", "housing","loan","contact"
,"month","poutcome")){
  a1[,i]<- as.numeric(a1[,i])
}
str(a1)

#visualization and exploratory analysis
library(corrplot)

bank<- a1
bank<- cbind(a1, a$y)
bank$`a$y`<-as.numeric(bank$`a$y`)
colnames(bank)[17]<-"class"
corr<- cor(bank)
round(corr,2)

corrplot(corr, method = "circle", number.cex = 0.55)

#Education level wise Yes/No
bar<-table(a$y, a$education)
barplot(bar, main="Education Level wise Yes/No", col=c("darkblue","red"),
    legend = rownames(bar), ylim = c(0,2500))

#prop.table(table(a$education, a$y))
```

```
#Month wise Yes/No
bar<-table (a$y, a$month)
barplot (bar, main="Month wise Yes/No"
  , col=c("darkblue","red"),
    legend = rownames(bar), ylim = c(0,1500), cex.names=.7, las = 2)

#prop. table(table(a$month, a$y))

#Job wise Yes/No
bar<-table (a$y, a$job)
barplot (bar, main="Job wise Yes/No"
  , col=c("darkblue","red"),
    legend = rownames(bar), ylim = c(0,1000), cex.names=.7, las = 2)

#prop.table(table(a$job, a$y))

# contact typewise Yes/No

bar<-table(a$y, a$contact)
barplot(bar, main="contact typewise Yes/No"
  , col=c("darkblue","red"),
    legend = rownames(bar), ylim = c(0,3000), cex.names=.7, las = 1)

#prop.table(table(a$contact, a$y))

#Data Partition for modelbuilding
set.seed(12)
trainrow<- sample(1:nrow(a1), 0.7*nrow(a1))
train<- a1[trainrow, ]
test<- a1[-trainrow, ]
str(train)

str(test)

#capturing labels of training and test data set
train.labels<- a[1:3164,17]
test.labels<- a[3165:4521,17]

prop.table(table(train.labels))

prop.table(table(test.labels))


#elbow test for deciding the number of K for K-NN model
plot.wgss = function(mydata, maxc) {
wss = numeric(maxc)
for (i in 1:maxc) wss[i] = kmeans(mydata,centers=i, nstart = 10)$tot.withinss
plot(1:maxc, wss, type="b", xlab="Number of Clusters",
ylab="Within groups sum of squares", main="Scree Plot") }
plot.wgss(a1, 10) # Elbow test
```

```r
#Building the K-NN model and confusion matrix
library(class)

pred1<- knn(train = train, test = test, cl = train.labels, k = 3)
#summary(pred1)
library(caret)

confusionMatrix(pred1,test.labels)

#cor(as.numeric(test.labels), as.numeric(pred1))

#modifying data for further analysis/models
a2<- cbind(a1,a$y)
colnames(a2)[17]<-"convert"
#str(a2)

#data partition for train and test
library(caret)
library(e1071)

set.seed(5)
partition<-createDataPartition(y = a2$convert, p=0.7, list= F)
train<- a2[partition,]
test<- a2[-partition,]
#str(train)
#str(test)

#Logistic regression taking all variables
model <- glm(convert ~., data = train, family = binomial(logit))
summary(model)

predscore<- predict(model, test, type = "response")
#head(predscore)

k<-cbind(test,predscore)
clasify <- array(c(99))
for(i in 1:nrow(k)){
  if(k[i,18]<0.3){
    clasify[i]<-"no"
  } else{
    clasify[i]<-"yes"
  }
}

confusionMatrix(clasify,(k$convert))

#cor(as.numeric(k$convert),as.numeric(as.factor(clasify)))

#summary(model)

#Logistic regression taking significant and high weight variables
model1<-glm(convert ~ age  + education +  housing + loan + contact + duration
+ campaign + pdays + previous + poutcome + month , data = train, family = bin
omial(logit))
summary(model1)

predscore1<- predict(model1, test, type = "response")
#head(predscore1)
k1<-cbind(test,predscore1)
clasify1 <- array(c(99))
for(i in 1:nrow(k1)){
  if(k1[i,18]<0.3){
    clasify1[i]<-"no"
  } else{
    clasify1[i]<-"yes"
  }
}

confusionMatrix(clasify1,k1$convert)

#cor(as.numeric(k1$convert),as.numeric(as.factor(clasify1)))

#Logistic regression taking only significant variables
sigmodel<-glm(convert ~ age  +  housing + loan + contact + duration + campaig
n + pdays + previous , data = train, family = binomial(logit))
#summary(sigmodel)

sigpredscore<- predict(sigmodel, test, type = "response")
#head(predscore1)
ks<-cbind(test,sigpredscore)
sclasify <- array(c(99))
for(i in 1:nrow(ks)){
  if(ks[i,18]<0.3){
    sclasify[i]<-"no"
  } else{
    sclasify[i]<-"yes"
  }
}

confusionMatrix(sclasify,ks$convert)

#cor(as.numeric(ks$convert),as.numeric(as.factor(sclasify)))

#Linear discriminant analysis/model and confusion matrix
library(MASS)

model2<- lda(convert ~., data = train)
#model2

pred2<- predict(model2, test)$class

table(pred2,test$convert)
```

```
confusionMatrix(pred2,test$convert)

#cor(as.numeric(pred2),as.numeric(test$convert))

#support vector machine model and confusion matrix
#library(e1071)
library(kernlab)

svmodel<- ksvm(convert ~ ., train, kernel = "rbfdot", gamma = 13)
#svmodel1<- svm(convert ~ ., train, kernel = "sigmoid")
#summary(svmodel)
pred3<- predict(svmodel,test)
confusionMatrix(pred3, test$convert)

#cor(as.numeric(pred3), as.numeric(test$convert))

#Decision tree model using c50 and confusion matrix
library(C50)

model3<- C5.0(train[-17], train$convert, trials = 11)
pred4<- predict(model3,test, type = "class")
confusionMatrix((pred4),test$convert)

#cor(as.numeric(pred4), as.numeric(test$convert))

#summary(model3)

#ddecision tree using rpart and confusion matrix
library(rpart)
model4<- rpart(convert ~., data = train, method = "class")

pred5 <- predict(model4, test, type  = "class")

confusionMatrix(pred5,test$convert)

#cor(as.numeric(test$convert), as.numeric(pred5))

#data processing for applying neural network concept
library(neuralnet)

a3<-a2
#str(a3)
a3$convert<- as.numeric(a3$convert)
set.seed(5)
trainrow<- sample(1:nrow(a3), 0.7*nrow(a3))
ntrain<- a3[trainrow, ]
ntest<- a3[-trainrow, ]

#neural network model, prediction, and confusion matrix


net.a3 <-neuralnet(convert ~ age + job + marital + education + default + bala
nce + housing + loan + contact + day + month + duration + campaign + pdays +
previous + poutcome, data = ntrain, hidden = 3, learningrate.limit = NULL, le
arningrate = 0.00001)

#plot(net.a3)
model_results <-compute(net.a3, ntest[1:16])

predict_convert<- model_results$net.result

nclasify <- array(c(99))
for(i in 1:nrow(predict_convert)){
  if(predict_convert[i,1]<1.5){

    nclasify[i]<-"1"
  } else{
    nclasify[i]<-"2"
  }
}

confusionMatrix(nclasify, ntest$convert)
```

2.Decision Tree Model (c50)

```
-----  Trial 10:  -----

Decision tree:

duration <= -0.3231062: no (462.5/30.4)
duration > -0.3231062:
:...duration > 1.958921:
    :...marital <= 1: yes (60.8/11.1)
    :   marital > 1:
    :   :...previous > 0.2700939: yes (24.9/4.2)
    :       previous <= 0.2700939:
    :       :...job <= 5:
    :           :...day <= 15: no (95.8/9.3)
    :           :   day > 15: yes (145.6/67.3)
    :           job > 5:
    :           :...education <= 1: yes (28.1)
    :               education > 1:
    :               :...pdays <= 0.8413145: yes (152.7/54.3)
    :                   pdays > 0.8413145: no (11.3)
    duration <= 1.958921:
    :...contact > 2: no (222.9/22.1)
        contact <= 2:
        :...month > 10:
            :...housing > 1: yes (24.7)
            :   housing <= 1:
            :   :...duration <= -0.111451: no (40.8)
            :       duration > -0.111451: yes (122.6/54.9)
            month <= 10:
            :...balance > 0.06291194:
                :...previous > 0.2700939:
                :   :...balance <= 2.144225: yes (159.1/49.4)
```

```
                  :   :   balance > 2.144225: no (18.3/1.7)
                  :   previous <= 0.2700939:
                  :   :...education > 3: yes (22.3/4.4)
                  :       education <= 3:
                  :       :...month > 8: no (111.7/10.2)
                  :           month <= 8:
                  :           :...balance <= 0.1426557: no (24.4)
                  :               balance > 0.1426557:
                  :               :...marital <= 1: no (28.8/4.9)
                  :                   marital > 1:
                  :                   :...balance <= 0.1605981: yes (12.7/0.2)
                  :                       balance > 0.1605981:
                  :                       :...balance <= 0.3390249: no (24.3)
                  :                           balance > 0.3390249:
                  :                           :...balance <= 0.6473676: yes (57.6/10.5)
                  :                               balance > 0.6473676: no (136.5/47.8)
                  balance <= 0.06291194:
                  :...duration <= -0.1691752: no (92.6)
                      duration > -0.1691752:
                      :...previous > 3.22245: no (30.3/1.5)
                          previous <= 3.22245:
                          :...age > 1.874954: yes (73/35.3)
                              age <= 1.874954:
                              :...education > 3: no (27.7/2.1)
                                  education <= 3:
                                  :...duration <= 0.2271972: no (220.3/31.4)
                                      duration > 0.2271972:
                                      :...pdays > 2.189681: yes (87.5/41.8)
                                          pdays <= 2.189681:
                                          :...loan > 1: no (38.2)
                                              loan <= 1:
                                              :...poutcome <= 1: no (43.4)
                                                  poutcome > 1:
                                                  :...day <= 4: yes (52.1/15.2)
                                                      day > 4:
                                                      :...age > 1.68585: no (17.4)
                                                          age <= 1.68585:
                                                          :...age <= 0.8348836: no (330.3/87)
                                                              age > 0.8348836: yes (53.8/18.7)
```