CKME - BANK MARKETING DATASET

Installation of Packages and adding it to library

install.packages("ggplot2") install.packages("corrplot") install.packages("caret") install.packages("dplyr") install.packages("caTools") install.packages("faraway") install.packages("modelr") install.packages("ROCR") install.packages("randomForest") install.packages("ROSE")

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caTools)
library(faraway)
```

```
##
## Attaching package: 'faraway'

## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
library(modelr)
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

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

```
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```r
library(e1071)
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

SET WORKING DIRECTORY

```r
setwd("C:/Users/iss/Desktop/Ryerson/CKME136/BANK")
```

READ CSV FILE

Dataset is obtained from UCI.edu related to European banks marketing campaign carried for term deposits.

```r
bank = read.csv("bank-additional-full.csv",sep=";", header=T)
```

SUMMARY OF THE DATASET

```r
summary(bank)
```

```
##       age                  job            marital     
##  Min.   :17.00   admin.     :10422   divorced: 4612  
##  1st Qu.:32.00   blue-collar: 9254   married :24928  
##  Median :38.00   technician : 6743   single  :11568  
##  Mean   :40.02   services   : 3969   unknown :   80  
##  3rd Qu.:47.00   management : 2924                   
##  Max.   :98.00   retired    : 1720                   
##                  (Other)    : 6156                   
##                education        default         housing     
##  university.degree  :12168   no     :32588   no     :18622  
##  high.school        : 9515   unknown: 8597   unknown:  990  
##  basic.9y           : 6045   yes    :    3   yes    :21576  
##  professional.course: 5243                                  
##  basic.4y           : 4176                                  
##  basic.6y           : 2292                                  
##  (Other)            : 1749                                  
##       loan            contact          month       day_of_week
##  no     :33950   cellular :26144   may    :13769   fri:7827   
##  unknown:  990   telephone:15044   jul    : 7174   mon:8514   
##  yes    : 6248                     aug    : 6178   thu:8623   
##                                    jun    : 5318   tue:8090   
##                                    nov    : 4101   wed:8134   
##                                    apr    : 2632              
##                                    (Other): 2016              
##     duration         campaign          pdays          previous    
##  Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.000  
##  1st Qu.: 102.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.000  
##  Median : 180.0   Median : 2.000   Median :999.0   Median :0.000  
##  Mean   : 258.3   Mean   : 2.568   Mean   :962.5   Mean   :0.173  
##  3rd Qu.: 319.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.000  
##  Max.   :4918.0   Max.   :56.000   Max.   :999.0   Max.   :7.000  
##                                                                   
##         poutcome      emp.var.rate      cons.price.idx  cons.conf.idx  
##  failure    : 4252   Min.   :-3.40000   Min.   :92.20   Min.   :-50.8  
##  nonexistent:35563   1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7  
##  success    : 1373   Median : 1.10000   Median :93.75   Median :-41.8  
##                      Mean   : 0.08189   Mean   :93.58   Mean   :-40.5  
##                      3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4  
##                      Max.   : 1.40000   Max.   :94.77   Max.   :-26.9  
##                                                                        
##    euribor3m      nr.employed     y        
##  Min.   :0.634   Min.   :4964   no :36548  
##  1st Qu.:1.344   1st Qu.:5099   yes: 4640  
##  Median :4.857   Median :5191              
##  Mean   :3.621   Mean   :5167              
##  3rd Qu.:4.961   3rd Qu.:5228              
##  Max.   :5.045   Max.   :5228              
##                                            
```

```r
head(bank)
```

```
##   age       job marital   education default housing loan   contact month
```

```
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
## 3         93.994         -36.4     4.857        5191 no
## 4         93.994         -36.4     4.857        5191 no
## 5         93.994         -36.4     4.857        5191 no
## 6         93.994         -36.4     4.857        5191 no
```

STRUCTURE OF THE DATASET

There are total of 21 columns and 41,118 observations in the dataset. 10 variables are numeric and 11 variables are characters including target variable that is the outcome of the call.

```r
str(bank)
```

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

OUTCOME VARIABLE - y

It is categorical variable, Yes representing that client has subscribed a term deposit? The data is considered to be imbalanced due a vast difference in class of outcome variable, there are 11% records for yes i.e subscribe for term deposit and 89% for not interested customers.

Outcome response category is converted to Binary

```
table (bank$y)
```

```
##
##    no   yes
## 36548  4640
```

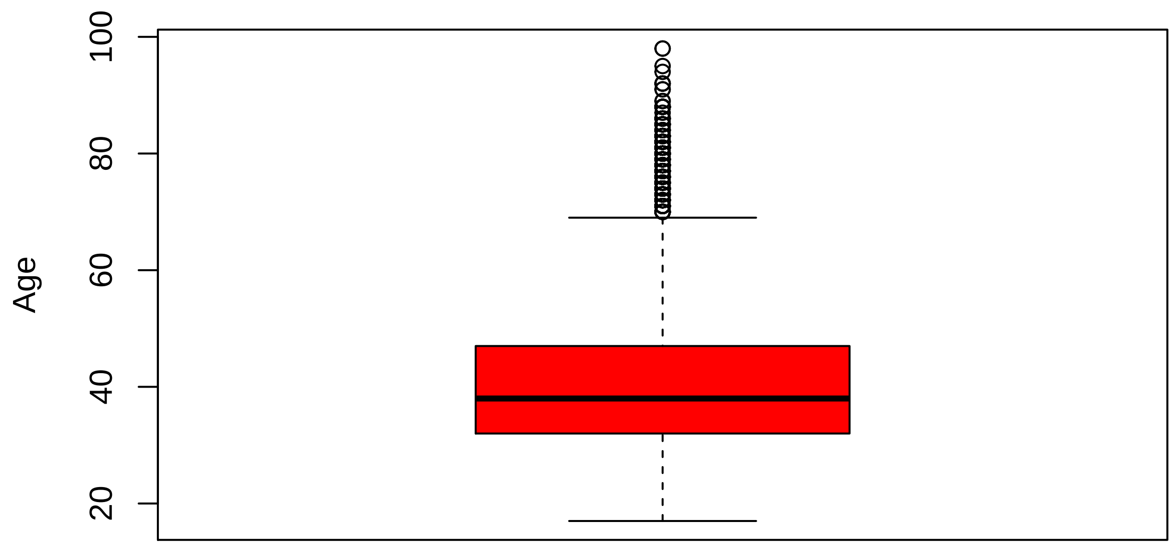
```
bank$y <- ifelse( bank$y == "yes", 1, 0)
table (bank$y)
```

```
##
##     0     1
## 36548  4640
```

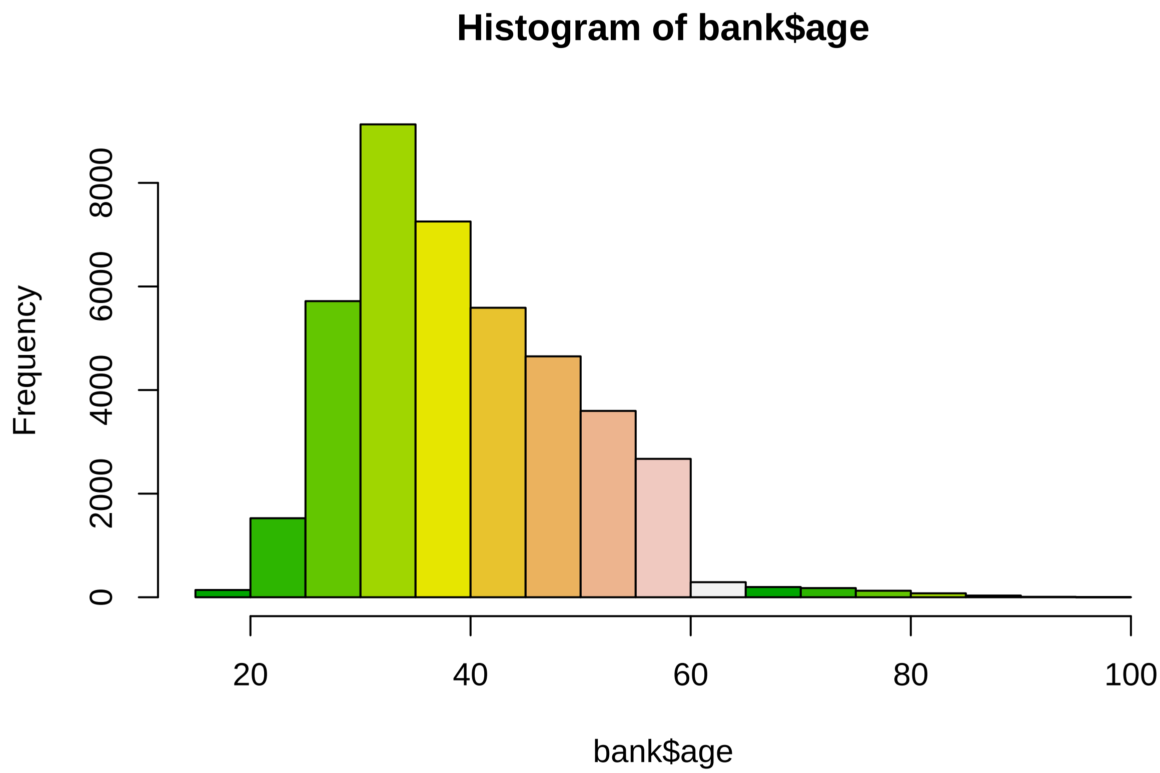
AGE VARIABLE

There are no such outliers in age variable. Majority of the records have age 60 or below. Looking at the boxplot and distribution of age variable, variable is converted into four different categories. After converting category of age variable to numeric actuall variable will be removed and age_4 is removed to keep n-1 in one hot encoding.

```
boxplot(bank$age, xlab="", ylab="Age",vertical=TRUE,col=2)
```




```
hist(bank$age,col=terrain.colors(10))
```




```
mean(bank$age)
```

```
## [1] 40.02406
```

```
median(bank$age)
```

```
## [1] 38
```

```
max(bank$age)
```

```
## [1] 98
```

```
min(bank$age)
```

```
## [1] 17
```

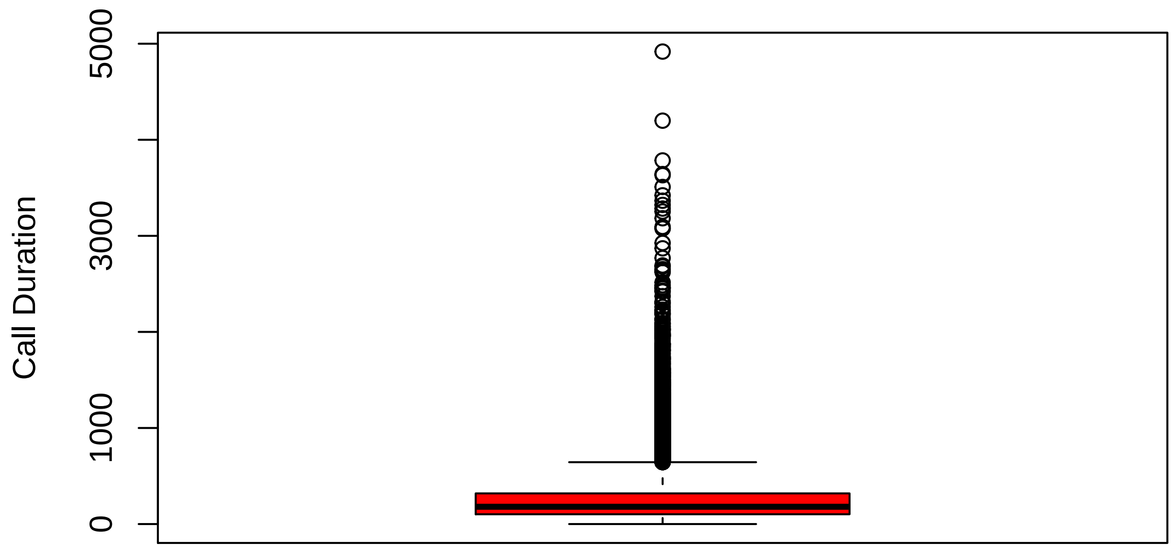
```
bank$age_1 <- as.numeric(bank$age <= 30)
bank$age_2 <- as.numeric(bank$age > 30 & bank$age <= 45)
bank$age_3 <- as.numeric(bank$age > 45 & bank$age <= 60)
bank$age_4 <- as.numeric(bank$age > 60)
```

DURATION VARIABLE

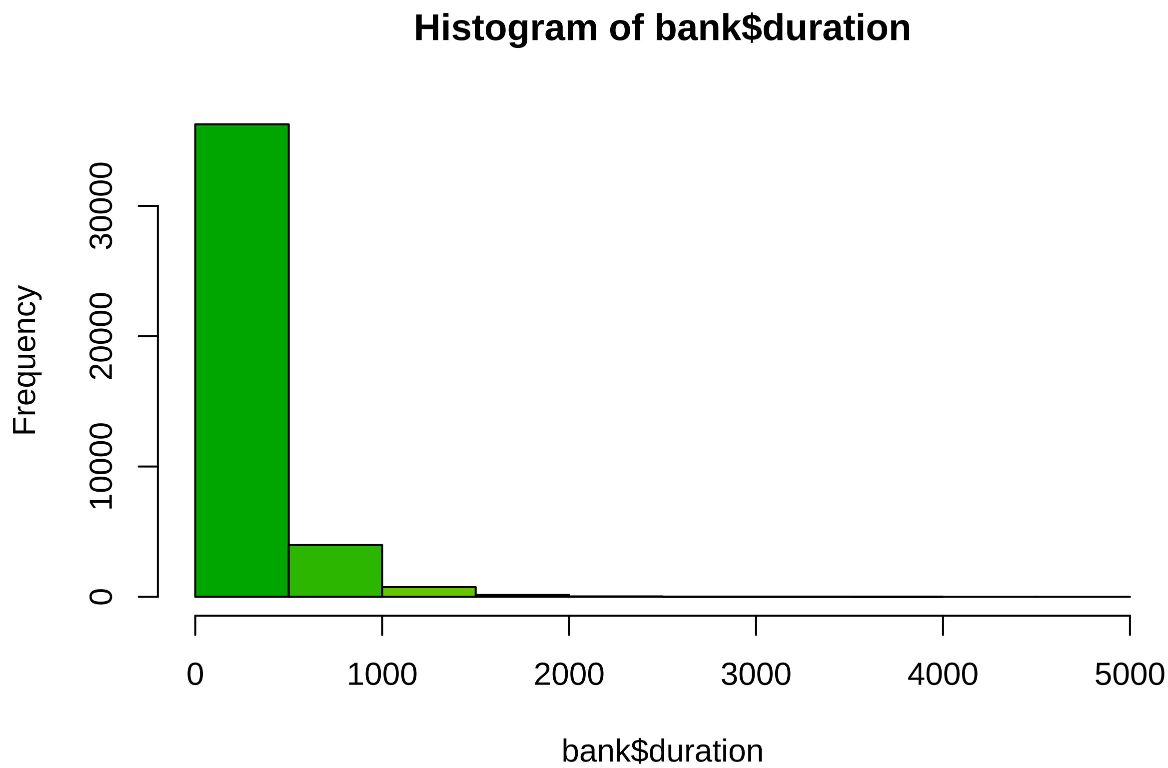
Duration represents the duration of last contact to the customer.

Variable is highly correlated with the outcome variable. However, to keep the model more realistic this variable will be removed since this could only be known when the call is actually made to the customer.

```
boxplot(bank$duration, xlab="", ylab="Call Duration",vertical=TRUE,col=2)
```



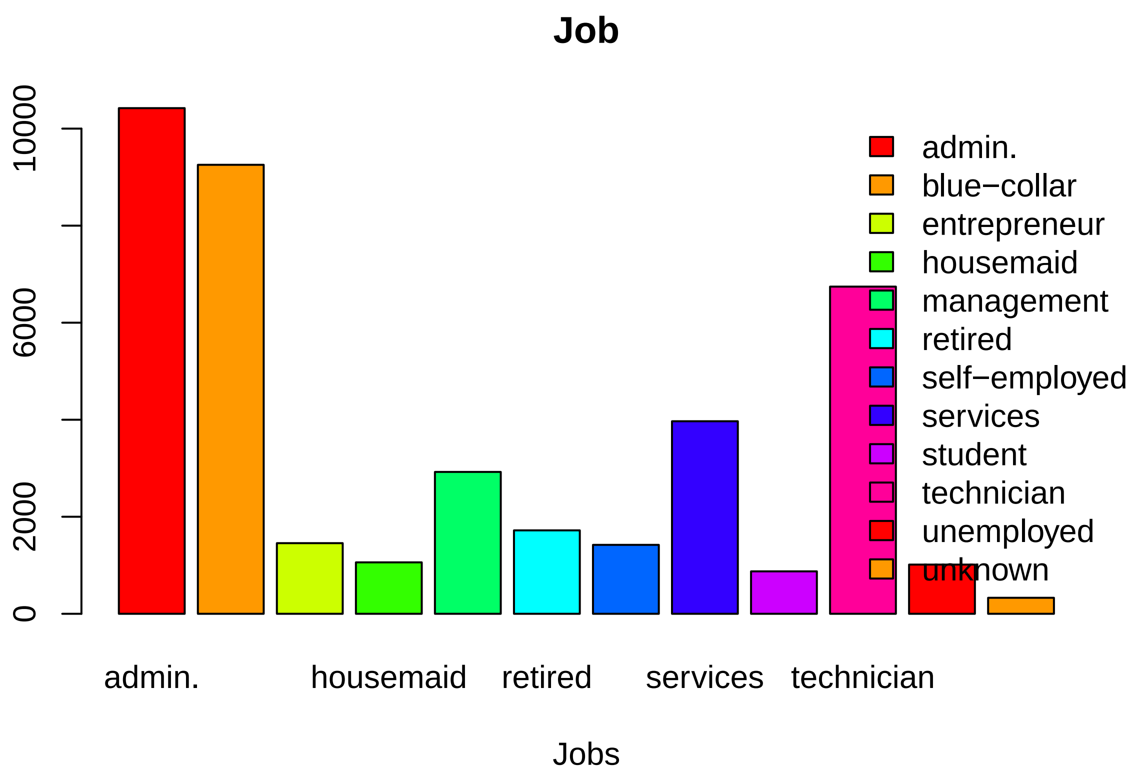

```
hist(bank$duration,col=terrain.colors(10))
```



Histogram of bank$duration
Frequency
0
10000
20000
30000
0
1000
2000
3000
4000
5000
bank$duration

JOB VARIABLE

```
barplot(table(bank$job), main="Job", xlab="Jobs",col=rainbow(10),legend.text = TRUE, beside=FALSE,args.l
```




```
table (bank$job, bank$y)
```

```
##
##                   0    1
##   admin.       9070 1352
##   blue-collar  8616  638
##   entrepreneur 1332  124
##   housemaid     954  106
##   management   2596  328
##   retired      1286  434
##   self-employed 1272  149
##   services     3646  323
##   student       600  275
##   technician   6013  730
##   unemployed    870  144
##   unknown       293   37
```

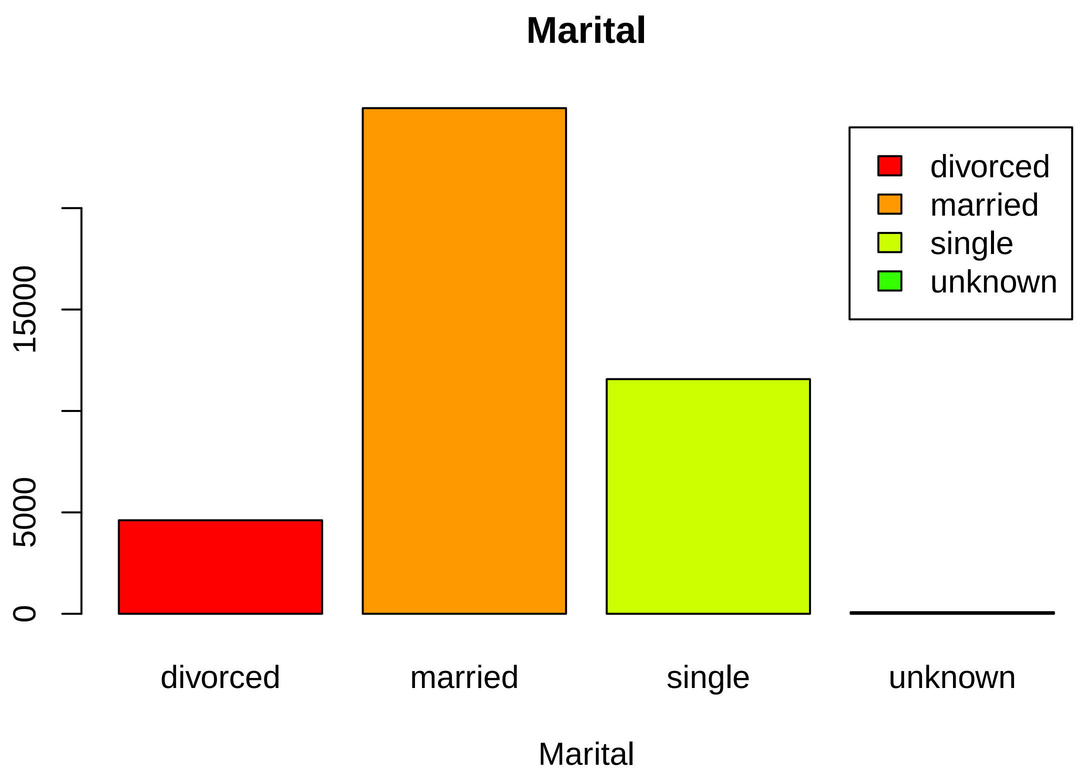
```
chisq.test(bank$job, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$job and bank$y
```

```
## X-squared = 961.24, df = 11, p-value < 2.2e-16
```

MARITAL VARIABLE

```
barplot(table(bank$marital), main="Marital", xlab="Marital", col=rainbow(10),legend.text = TRUE, beside=
```




```
table (bank$marital, bank$y)
```

```
##
##                0     1
##   divorced  4136   476
##   married  22396  2532
##   single    9948  1620
##   unknown     68    12
```

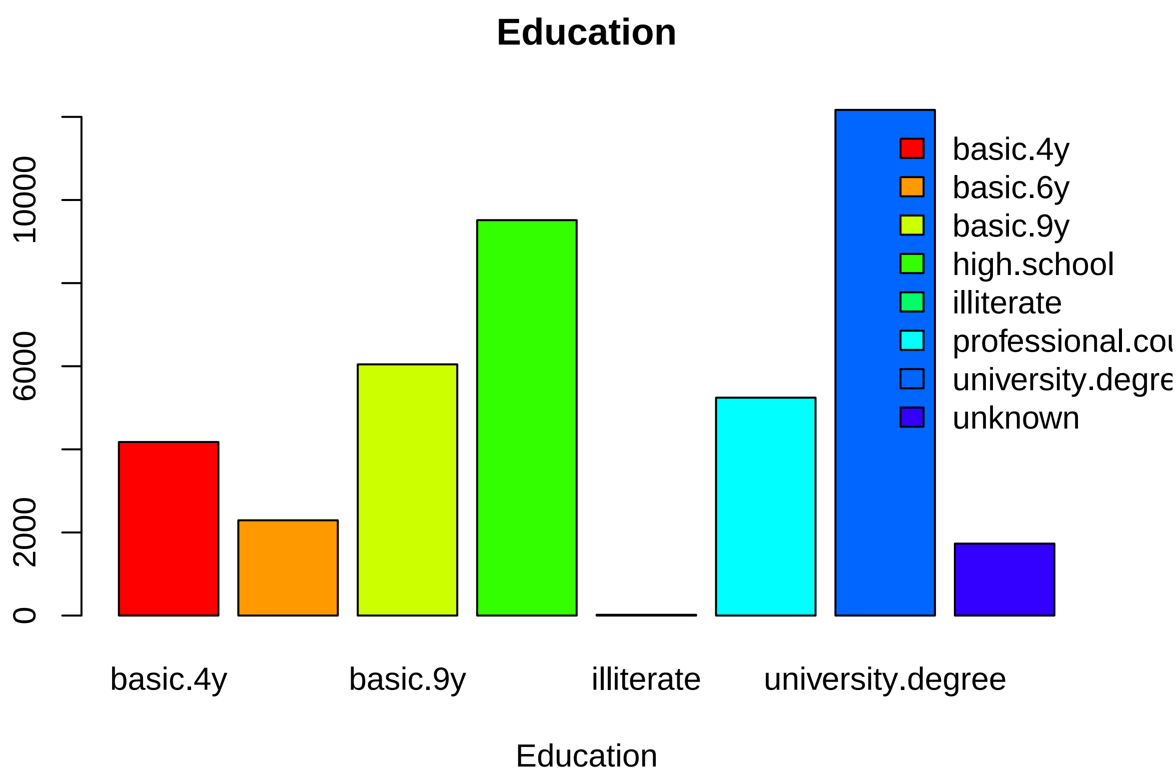
```
chisq.test(bank$marital, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$marital and bank$y
## X-squared = 122.66, df = 3, p-value < 2.2e-16
```

EDUCATION VARIABLE

In Education variable, class illiterate had very less instances. With the class 'Illiterate' R suggested that the CHI Sq approximation may be incorrect. Therefore the class was removed.

```
barplot(table(bank$education), main="Education", xlab="Education", col=rainbow(10),legend.text = TRUE,,a
```




```
table (bank$education, bank$y)
```

```
##
##                          0     1
##   basic.4y            3748   428
##   basic.6y            2104   188
##   basic.9y            5572   473
##   high.school         8484  1031
##   illiterate            14     4
##   professional.course 4648   595
##   university.degree  10498  1670
##   unknown             1480   251
```

```
chisq.test(bank$education, bank$y, correct=FALSE)
```

```
## Warning in chisq.test(bank$education, bank$y, correct = FALSE): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$education and bank$y
```

```
## X-squared = 193.11, df = 7, p-value < 2.2e-16
```

```
bank <- bank %>% filter(education != "illiterate")
chisq.test(bank$education, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$education and bank$y
## X-squared = 191.01, df = 6, p-value < 2.2e-16
```

DEFAULT VARIABLE

default variable explains if the client has default on credit products. Class 'yes' has very low records therefore the records will be excluded.

```
table (bank$default, bank$y)
```

```
##
##              0     1
##   no      28383  4194
##   unknown  8148   442
##   yes        3     0
```

```
chisq.test(bank$default, bank$y, correct=FALSE)
```

```
## Warning in chisq.test(bank$default, bank$y, correct = FALSE): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$default and bank$y
## X-squared = 406.71, df = 2, p-value < 2.2e-16
```

```
bank <- bank %>% filter(default != "yes")
chisq.test(bank$default, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$default and bank$y
## X-squared = 406.3, df = 1, p-value < 2.2e-16
```

HOUSING VARIABLE

Housing represents if the customer have any House loan. The Chi Square value 0.05 which can be consider at the border of 95% significance value.

```r
table (bank$housing, bank$y)
```

```
##
##                0     1
##   no       16587  2025
##   unknown    883   107
##   yes      19061  2504
```

```r
chisq.test(bank$housing, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$housing and bank$y
## X-squared = 5.5553, df = 2, p-value = 0.06218
```

LOAN VARIABLE

Loan represents if the customer have any personal loan. The Chi Square value is 0.57 which shows the loan doesnt have a signifance on the outcome variable Y. Therefore, this variable will be removed.

```r
table (bank$loan, bank$y)
```

```
##
##                 0     1
##   no        30085  3847
##   unknown     883   107
##   yes        5563   682
```

```r
chisq.test(bank$loan, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$loan and bank$y
## X-squared = 1.1248, df = 2, p-value = 0.5698
```

```r
table (bank$contact, bank$y)
```

```
##
##                 0     1
##   cellular  22276  3850
##   telephone 14255   786
```

```r
chisq.test(bank$contact, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$contact and bank$y
## X-squared = 863.98, df = 1, p-value < 2.2e-16
```

```r
table (bank$day_of_week, bank$y)
```

```
##
##           0    1
##   fri 6977  846
##   mon 7666  847
##   thu 7574 1043
##   tue 7130  952
##   wed 7184  948
```

```r
chisq.test(bank$day_of_week, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$day_of_week and bank$y
## X-squared = 25.795, df = 4, p-value = 3.48e-05
```

```r
table (bank$poutcome, bank$y)
```

```
##
##                   0     1
##   failure      3645   605
##   nonexistent 32407  3138
##   success       479   893
```

```r
chisq.test(bank$poutcome, bank$y, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bank$poutcome and bank$y
## X-squared = 4225.9, df = 2, p-value < 2.2e-16
```

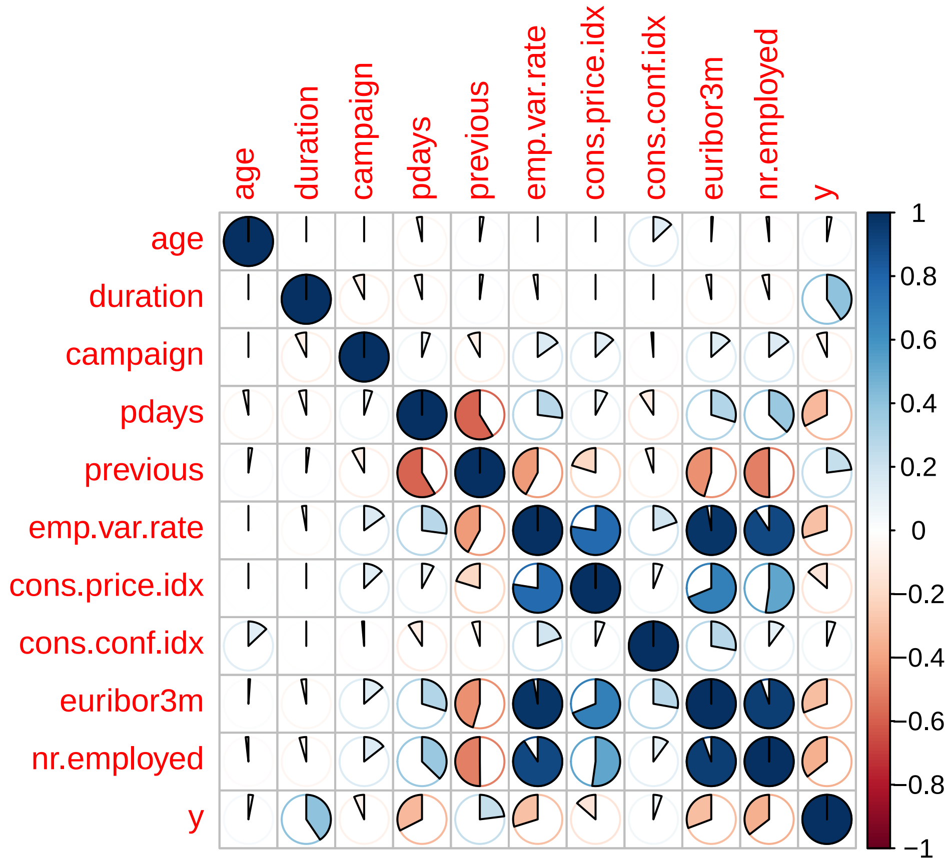
CORRELATION MATRIX

Checking Correlation for all the numeric variables. Strong correlation is observed for all economic variables emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, and nr.employed.

```
round(cor(bank[,c(1,11:14,16:21)]),2)
```

```
##                  age duration campaign pdays previous emp.var.rate
## age             1.00     0.00     0.00 -0.03     0.02         0.00
## duration        0.00     1.00    -0.07 -0.05     0.02        -0.03
## campaign        0.00    -0.07     1.00  0.05    -0.08         0.15
## pdays          -0.03    -0.05     0.05  1.00    -0.59         0.27
## previous        0.02     0.02    -0.08 -0.59     1.00        -0.42
## emp.var.rate    0.00    -0.03     0.15  0.27    -0.42         1.00
## cons.price.idx  0.00     0.01     0.13  0.08    -0.20         0.78
## cons.conf.idx   0.13    -0.01    -0.01 -0.09    -0.05         0.20
## euribor3m       0.01    -0.03     0.14  0.30    -0.45         0.97
## nr.employed    -0.02    -0.04     0.14  0.37    -0.50         0.91
## y               0.03     0.41    -0.07 -0.32     0.23        -0.30
##                cons.price.idx cons.conf.idx euribor3m nr.employed     y
## age                      0.00          0.13      0.01       -0.02  0.03
## duration                 0.01         -0.01     -0.03       -0.04  0.41
## campaign                 0.13         -0.01      0.14        0.14 -0.07
## pdays                    0.08         -0.09      0.30        0.37 -0.32
## previous                -0.20         -0.05     -0.45       -0.50  0.23
## emp.var.rate             0.78          0.20      0.97        0.91 -0.30
## cons.price.idx           1.00          0.06      0.69        0.52 -0.14
## cons.conf.idx            0.06          1.00      0.28        0.10  0.05
## euribor3m                0.69          0.28      1.00        0.95 -0.31
## nr.employed              0.52          0.10      0.95        1.00 -0.35
## y                       -0.14          0.05     -0.31       -0.35  1.00
```

```
corrplot(cor(bank[,c(1,11:14,16:21)]), method = "pie")
```


age
duration
campaign
pdays
previous
emp.var.rate
cons.price.idx
cons.conf.idx
euribor3m
nr.employed
y
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

#VIF To avoid multicolinearity, Variance Inflation Factor was checked for different group of variables Social & Economic and Campaign. Value of VIF seems to be really high therefore i have considered to remove the economic variables and only keep "euribor3m" and "cons.conf.idx"

For campaign related variables, duration is highly correlated to outcome variable but this could only be known when we make the call, so i will exclude from the analysis.

pdays and previous are the variables related previous contact. pdays have high number of no contact value '999' therefore i will remove pdays from the model.

Used library "faraway" to use the function of "vif"

```
mymodel_eco <- glm(y ~ emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m + nr.employed ,data=ba

mymodel_cam <- glm(y ~ duration + pdays + previous,data=bank, family=binomial)

summary(mymodel_eco)
```

```
##
## Call:
## glm(formula = y ~ emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed, family = binomial, data = bank)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1759  -0.3703  -0.3388  -0.2658   2.6193
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -21.784535  13.084147  -1.665   0.0959 .
## emp.var.rate    -0.490054   0.057568  -8.513  < 2e-16 ***
## cons.price.idx   0.628745   0.083349   7.543 4.58e-14 ***
## cons.conf.idx    0.034177   0.005016   6.814 9.50e-12 ***
## euribor3m        0.053554   0.072358   0.740   0.4592
## nr.employed     -0.007404   0.001218  -6.079 1.21e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28977  on 41166  degrees of freedom
## Residual deviance: 24329  on 41161  degrees of freedom
## AIC: 24341
##
## Number of Fisher Scoring iterations: 5
```

```
summary(mymodel_cam)
```

```
##
## Call:
## glm(formula = y ~ duration + pdays + previous, family = binomial,
##     data = bank)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.5561  -0.3778  -0.2987  -0.2555   2.6781
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.178e+00  8.390e-02  -14.05   <2e-16 ***
## duration     3.891e-03  6.206e-05   62.70   <2e-16 ***
## pdays       -2.526e-03  8.034e-05  -31.44   <2e-16 ***
## previous     4.053e-01  3.710e-02   10.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28977  on 41166  degrees of freedom
## Residual deviance: 21345  on 41163  degrees of freedom
## AIC: 21353
##
## Number of Fisher Scoring iterations: 5
```

```r
vif(mymodel_eco)
```

```
##   emp.var.rate cons.price.idx  cons.conf.idx      euribor3m    nr.employed
##      336.68210       95.81285       22.18179      648.41374      318.87454
```

```r
vif(mymodel_cam)
```

```
##  duration     pdays  previous
## 10.659277  9.281471 13.879066
```

Considering the correlation, CHI Sq and multicolinearity between all the variables following variables will not be considered in the model.

```r
bank$pdays <- NULL
bank$emp.var.rate <- NULL
bank$cons.price.idx <- NULL
bank$nr.employed <- NULL
bank$loan <- NULL
bank$duration <- NULL
str(bank)
```

```
## 'data.frame':    41167 obs. of  19 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...
##  $ marital       : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ education     : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...
##  $ default       : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ housing       : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ contact       : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
##  $ month         : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ day_of_week   : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ cons.conf.idx: num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ y             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ age_1         : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ age_2         : num  0 0 1 1 0 1 0 1 0 0 ...
##  $ age_3         : num  1 1 0 0 1 0 1 0 0 0 ...
```

```
##  $ age_4         : num  0 0 0 0 0 0 0 0 0 0 ...
```

ONE HOT ENCODING

To convert categorical variable to numeric one hot encoding is considered. Since one hot encoding is n-1 categories we will delete one converted variable. The category/variable removed from the bank dataset is based on the ascending alphabetical order. Example: admin. from the job is first in order therefore job_admin. will be removed.

Variable job_blue-collar and job_self-employed has "-" in the name which will be renamed.

```r
for(LEVEL in unique(bank$job)){
  bank[paste("job", LEVEL, sep = "_")] <- ifelse(bank$job== LEVEL, 1, 0)}

bank <- bank %>% rename(job_blue_collar = `job_blue-collar`)
bank <- bank %>% rename(job_self_employed  = `job_self-employed`)

for(LEVEL in unique(bank$marital)){
  bank[paste("marital", LEVEL, sep = "_")] <- ifelse(bank$marital== LEVEL, 1, 0)}

for(LEVEL in unique(bank$education)){
  bank[paste("education", LEVEL, sep = "_")] <- ifelse(bank$education == LEVEL, 1, 0)}

for(LEVEL in unique(bank$default)){
  bank[paste("default", LEVEL, sep = "_")] <- ifelse(bank$default == LEVEL, 1, 0)}

for(LEVEL in unique(bank$housing)){
  bank[paste("housing", LEVEL, sep = "_")] <- ifelse(bank$housing == LEVEL, 1, 0)}

for(LEVEL in unique(bank$contact)){
  bank[paste("contact", LEVEL, sep = "_")] <- ifelse(bank$contact == LEVEL, 1, 0)}

for(LEVEL in unique(bank$month)){
  bank[paste("month", LEVEL, sep = "_")] <- ifelse(bank$month == LEVEL, 1, 0)}

for(LEVEL in unique(bank$day_of_week)){
  bank[paste("day_of_week", LEVEL, sep = "_")] <- ifelse(bank$day_of_week == LEVEL, 1, 0)}

for(LEVEL in unique(bank$poutcome)){
  bank[paste("poutcome", LEVEL, sep = "_")] <- ifelse(bank$poutcome == LEVEL, 1, 0)}

str(bank)
```

```
## 'data.frame':    41167 obs. of  67 variables:
##  $ age                          : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job                          : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8
##  $ marital                      : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ..
##  $ education                    : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 .
##  $ default                      : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ housing                      : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ contact                      : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
##  $ month                        : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ day_of_week                  : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ campaign                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ previous                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome                     : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2
##  $ cons.conf.idx                : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 .
```

```
##  $ euribor3m                   : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ y                           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ age_1                       : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ age_2                       : num  0 0 1 1 0 1 0 1 0 0 ...
##  $ age_3                       : num  1 1 0 0 1 0 1 0 0 0 ...
##  $ age_4                       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_housemaid               : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ job_services                : num  0 1 1 0 1 1 0 0 0 1 ...
##  $ job_admin.                  : num  0 0 0 1 0 0 1 0 0 0 ...
##  $ job_blue_collar             : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ job_technician              : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ job_retired                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_management              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_unemployed              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_self_employed           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_unknown                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_entrepreneur            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_student                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ marital_married             : num  1 1 1 1 1 1 1 1 0 0 ...
##  $ marital_single              : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ marital_divorced            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ marital_unknown             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ education_basic.4y          : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ education_high.school       : num  0 1 1 0 1 0 0 0 0 1 ...
##  $ education_basic.6y          : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ education_basic.9y          : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ education_professional.course: num  0 0 0 0 0 0 1 0 1 0 ...
##  $ education_unknown           : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ education_university.degree  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ default_no                  : num  1 0 1 1 1 0 1 0 1 1 ...
##  $ default_unknown             : num  0 1 0 0 0 1 0 1 0 0 ...
##  $ housing_no                  : num  1 1 0 1 1 1 1 1 0 0 ...
##  $ housing_yes                 : num  0 0 1 0 0 0 0 0 1 1 ...
##  $ housing_unknown             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ contact_telephone           : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_cellular            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_may                   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ month_jun                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_jul                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_aug                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_oct                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_nov                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_dec                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_mar                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_apr                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_sep                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_mon             : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ day_of_week_tue             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_wed             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_thu             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_fri             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome_nonexistent        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ poutcome_failure            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome_success            : num  0 0 0 0 0 0 0 0 0 0 ...
```

Since the factors are converted to numeric variable, original categorical variables will be removed.

```
bank$job <- NULL
bank$marital <- NULL
bank$education <- NULL
bank$default <- NULL
bank$housing <- NULL
bank$contact <- NULL
bank$month <- NULL
bank$day_of_week <- NULL
bank$poutcome <- NULL
bank$age <-NULL
bank$default_yes <- NULL
bank$education_illiterate <- NULL
bank$job_admin. <- NULL
bank$marital_divorced <- NULL
bank$education_basic.4y <- NULL
bank$default_no <- NULL
bank$housing_no <- NULL
bank$contact_cellular <- NULL
bank$month_apr <- NULL
bank$day_of_week_fri <- NULL
bank$poutcome_failure <- NULL
bank$age_4 <- NULL

str(bank)
```

```
## 'data.frame':    41167 obs. of  47 variables:
##  $ campaign                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ previous                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ cons.conf.idx                : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 .
##  $ euribor3m                    : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ y                            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ age_1                        : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ age_2                        : num  0 0 1 1 0 1 0 1 0 0 ...
##  $ age_3                        : num  1 1 0 0 1 0 1 0 0 0 ...
##  $ job_housemaid                : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ job_services                 : num  0 1 1 0 1 1 0 0 0 1 ...
##  $ job_blue_collar              : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ job_technician               : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ job_retired                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_management               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_unemployed               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_self_employed            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_unknown                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_entrepreneur             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ job_student                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ marital_married              : num  1 1 1 1 1 1 1 1 0 0 ...
##  $ marital_single               : num  0 0 0 0 0 0 0 0 1 1 ...
##  $ marital_unknown              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ education_high.school        : num  0 1 1 0 1 0 0 0 0 1 ...
##  $ education_basic.6y           : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ education_basic.9y           : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ education_professional.course: num  0 0 0 0 0 0 1 0 1 0 ...
##  $ education_unknown            : num  0 0 0 0 0 0 0 1 0 0 ...
```

```
##  $ education_university.degree  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ default_unknown               : num  0 1 0 0 0 1 0 1 0 0 ...
##  $ housing_yes                   : num  0 0 1 0 0 0 0 0 1 1 ...
##  $ housing_unknown               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ contact_telephone             : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ month_may                     : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ month_jun                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_jul                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_aug                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_oct                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_nov                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_dec                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_mar                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ month_sep                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_mon               : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ day_of_week_tue               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_wed               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ day_of_week_thu               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome_nonexistent          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ poutcome_success              : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
colnames(bank)
```

```
##  [1] "campaign"                      "previous"
##  [3] "cons.conf.idx"                 "euribor3m"
##  [5] "y"                             "age_1"
##  [7] "age_2"                         "age_3"
##  [9] "job_housemaid"                 "job_services"
## [11] "job_blue_collar"               "job_technician"
## [13] "job_retired"                   "job_management"
## [15] "job_unemployed"                "job_self_employed"
## [17] "job_unknown"                   "job_entrepreneur"
## [19] "job_student"                   "marital_married"
## [21] "marital_single"                "marital_unknown"
## [23] "education_high.school"         "education_basic.6y"
## [25] "education_basic.9y"            "education_professional.course"
## [27] "education_unknown"             "education_university.degree"
## [29] "default_unknown"               "housing_yes"
## [31] "housing_unknown"               "contact_telephone"
## [33] "month_may"                     "month_jun"
## [35] "month_jul"                     "month_aug"
## [37] "month_oct"                     "month_nov"
## [39] "month_dec"                     "month_mar"
## [41] "month_sep"                     "day_of_week_mon"
## [43] "day_of_week_tue"               "day_of_week_wed"
## [45] "day_of_week_thu"               "poutcome_nonexistent"
## [47] "poutcome_success"
```

```r
dim(bank)
```

```
## [1] 41167    47
```

Rearranging the variable to have outcome First and then all other variables.

```
bank <- bank[,c(5,1,2,3,4,6:47)]
colnames(bank)
```

```
##  [1] "y"                        "campaign"
##  [3] "previous"                 "cons.conf.idx"
##  [5] "euribor3m"                "age_1"
##  [7] "age_2"                    "age_3"
##  [9] "job_housemaid"            "job_services"
## [11] "job_blue_collar"          "job_technician"
## [13] "job_retired"              "job_management"
## [15] "job_unemployed"           "job_self_employed"
## [17] "job_unknown"              "job_entrepreneur"
## [19] "job_student"              "marital_married"
## [21] "marital_single"           "marital_unknown"
## [23] "education_high.school"    "education_basic.6y"
## [25] "education_basic.9y"       "education_professional.course"
## [27] "education_unknown"        "education_university.degree"
## [29] "default_unknown"          "housing_yes"
## [31] "housing_unknown"          "contact_telephone"
## [33] "month_may"                "month_jun"
## [35] "month_jul"                "month_aug"
## [37] "month_oct"                "month_nov"
## [39] "month_dec"                "month_mar"
## [41] "month_sep"                "day_of_week_mon"
## [43] "day_of_week_tue"          "day_of_week_wed"
## [45] "day_of_week_thu"          "poutcome_nonexistent"
## [47] "poutcome_success"
```

SPLIT DATASET TO TRAIN AND TEST

Splitting the dataset Bank into Training and Test set by the ratio of 80% and 20% respectively.

```
set.seed(123)

split <- sample.split(bank$y, SplitRatio = 0.80)

train <- subset(bank, split==TRUE)
test <- subset(bank, split==FALSE)

table(train$y)
```

```
##
##     0     1
## 29225  3709
```

```
table(test$y)
```

```
##
##    0    1
## 7306  927
```

FITTING LOGISTIC REGRESSION MODEL TO THE TRAINING DATASET

```
r_model_lr <- glm(formula = y ~ ., data=train, family=binomial)

summary(r_model_lr)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2204  -0.4170  -0.3352  -0.2361   3.1175
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   0.104614   0.279435   0.374 0.708123
## campaign                     -0.053867   0.010410  -5.175 2.28e-07 ***
## previous                      0.218916   0.057638   3.798 0.000146 ***
## cons.conf.idx                 0.026385   0.004853   5.437 5.41e-08 ***
## euribor3m                    -0.448894   0.016079 -27.919  < 2e-16 ***
## age_1                        -0.249787   0.130789  -1.910 0.056152 .
## age_2                        -0.355287   0.122692  -2.896 0.003782 **
## age_3                        -0.299847   0.117238  -2.558 0.010540 *
## job_housemaid                -0.157762   0.145405  -1.085 0.277928
## job_services                 -0.173527   0.082965  -2.092 0.036477 *
## job_blue_collar              -0.168498   0.075881  -2.221 0.026382 *
## job_technician               -0.063919   0.068590  -0.932 0.351388
## job_retired                   0.088267   0.114714   0.769 0.441623
## job_management               -0.133678   0.083731  -1.597 0.110374
```

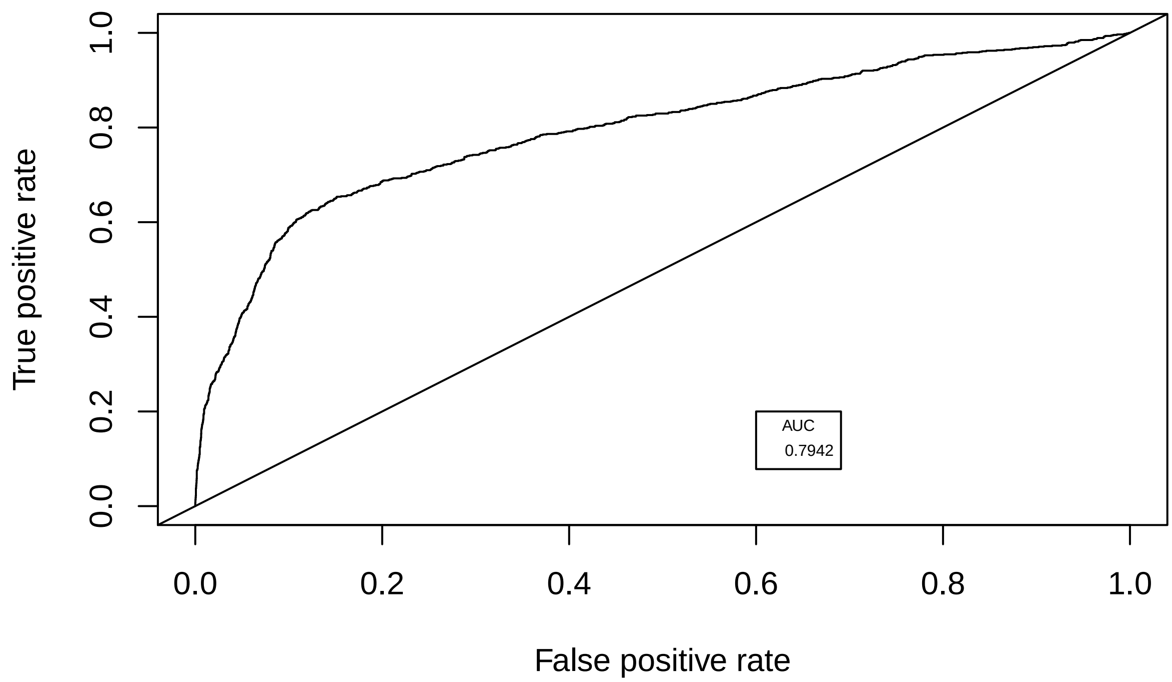
```
## job_unemployed                 0.040031   0.123421   0.324 0.745676
## job_self_employed             -0.086684   0.112094  -0.773 0.439340
## job_unknown                   -0.158908   0.236858  -0.671 0.502285
## job_entrepreneur              -0.113711   0.119302  -0.953 0.340521
## job_student                    0.244521   0.110834   2.206 0.027370 *
## marital_married                0.099951   0.066675   1.499 0.133853
## marital_single                 0.111403   0.075739   1.471 0.141327
## marital_unknown                0.411591   0.389487   1.057 0.290624
## education_high.school          0.071926   0.089085   0.807 0.419444
## education_basic.6y             0.095469   0.116618   0.819 0.412986
## education_basic.9y            -0.019052   0.091462  -0.208 0.834988
## education_professional.course  0.136688   0.098147   1.393 0.163712
## education_unknown              0.192143   0.116466   1.650 0.098989 .
## education_university.degree    0.134469   0.089267   1.506 0.131974
## default_unknown               -0.317455   0.064995  -4.884 1.04e-06 ***
## housing_yes                   -0.044689   0.039819  -1.122 0.261728
## housing_unknown               -0.049640   0.132283  -0.375 0.707471
## contact_telephone             -0.297827   0.060807  -4.898 9.69e-07 ***
## month_may                     -0.628745   0.072170  -8.712  < 2e-16 ***
## month_jun                      0.301918   0.088096   3.427 0.000610 ***
## month_jul                      0.355718   0.090093   3.948 7.87e-05 ***
## month_aug                     -0.079818   0.101351  -0.788 0.430963
## month_oct                      0.391376   0.123265   3.175 0.001498 **
## month_nov                     -0.104684   0.095816  -1.093 0.274590
## month_dec                      0.550288   0.197597   2.785 0.005354 **
## month_mar                      0.962451   0.119916   8.026 1.01e-15 ***
## month_sep                      0.119282   0.131341   0.908 0.363782
## day_of_week_mon               -0.150228   0.064132  -2.342 0.019157 *
## day_of_week_tue                0.109670   0.063293   1.733 0.083140 .
## day_of_week_wed                0.164230   0.063411   2.590 0.009599 **
## day_of_week_thu                0.102328   0.061750   1.657 0.097496 .
## poutcome_nonexistent           0.691920   0.095284   7.262 3.82e-13 ***
## poutcome_success               1.813838   0.086679  20.926  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23183  on 32933  degrees of freedom
## Residual deviance: 18480  on 32887  degrees of freedom
## AIC: 18574
##
## Number of Fisher Scoring iterations: 6
```

```
confusionMatrix(as.factor(ifelse(predict(r_model_lr, type="response",newdata=test) > 0.5,"1","0")), as.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7203  717
##          1  103  210
##
##                Accuracy : 0.9004
##                  95% CI : (0.8937, 0.9068)
```

```
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 8.033e-05
##
##                   Kappa : 0.2989
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.22654
##             Specificity : 0.98590
##          Pos Pred Value : 0.67093
##          Neg Pred Value : 0.90947
##              Prevalence : 0.11260
##          Detection Rate : 0.02551
##    Detection Prevalence : 0.03802
##       Balanced Accuracy : 0.60622
##
##        'Positive' Class : 1
##
```

```
r_pred_lr = predict(r_model_lr, type='response', newdata=test)

pred_lr_r<-prediction(r_pred_lr, test$y)
r_eval_lr <- performance(pred_lr_r,"tpr","fpr")
plot(r_eval_lr, colorize=F)
abline(a=0, b=1)

r_auc_lr <- performance(pred_lr_r,"auc")
r_auc_lr <- unlist(slot(r_auc_lr,"y.values"))
r_auc_lr <- round(r_auc_lr,4)
legend(.6,.2,r_auc_lr, title="AUC", cex=0.5)
```



Accuracy of the model is 90% but the sensitivity is very low that is prediciting the client who would sign up for the term deposit.

As mentioned earlier the dependent variable has imbalanced class, different re-sampling is conducted to improve the sensitivity as well as the accuracy.
Over sampling and Under sampling is done keeping the ratio of cal success/no-success to 70/30

SAME MODEL IS USED USING DIFFERENT SAMPLE SIZE TRAINING DATASET

```
over <- ovun.sample(y ~., data=train, method = "over", N=41750)$data
under <- ovun.sample(y ~., data=train, method = "under", N=12363)$data
both <- ovun.sample(y ~., data=train, method = "both", p=0.5, seed = 222, N=32934)$data
```

LOGISTIC REGRESSION - OVER SAMPLING

```
o_model_lr <- glm(formula = y ~ ., data=over, family=binomial)
summary(o_model_lr)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = over)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7915  -0.6842  -0.5194   0.6101   2.6654
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.603124   0.195196   8.213  < 2e-16 ***
## campaign                      -0.053069   0.006242  -8.502  < 2e-16 ***
## previous                       0.244864   0.046644   5.250 1.52e-07 ***
## cons.conf.idx                  0.033679   0.003398   9.913  < 2e-16 ***
## euribor3m                     -0.430420   0.010603 -40.595  < 2e-16 ***
## age_1                         -0.465715   0.091816  -5.072 3.93e-07 ***
## age_2                         -0.557972   0.086921  -6.419 1.37e-10 ***
## age_3                         -0.520492   0.083564  -6.229 4.70e-10 ***
## job_housemaid                 -0.114568   0.091916  -1.246  0.21260
## job_services                  -0.119769   0.051800  -2.312  0.02077 *
## job_blue_collar               -0.075975   0.047354  -1.604  0.10863
## job_technician                -0.073708   0.043966  -1.676  0.09365 .
## job_retired                    0.079827   0.076877   1.038  0.29910
## job_management                -0.099914   0.053670  -1.862  0.06266 .
## job_unemployed                -0.016266   0.083475  -0.195  0.84550
## job_self_employed             -0.123063   0.072580  -1.696  0.08997 .
## job_unknown                   -0.110844   0.159444  -0.695  0.48693
## job_entrepreneur              -0.132813   0.075610  -1.757  0.07899 .
## job_student                    0.324482   0.077341   4.195 2.72e-05 ***
## marital_married                0.056436   0.041853   1.348  0.17752
## marital_single                 0.091937   0.047455   1.937  0.05270 .
## marital_unknown                0.658450   0.241416   2.727  0.00638 **
## education_high.school          0.158348   0.057451   2.756  0.00585 **
## education_basic.6y             0.182273   0.072906   2.500  0.01241 *
## education_basic.9y             0.091429   0.057892   1.579  0.11427
## education_professional.course  0.303234   0.063263   4.793 1.64e-06 ***
## education_unknown              0.213194   0.076637   2.782  0.00540 **
## education_university.degree    0.233993   0.058120   4.026 5.67e-05 ***
## default_unknown               -0.324440   0.038874  -8.346  < 2e-16 ***
## housing_yes                   -0.025875   0.025546  -1.013  0.31112
## housing_unknown               -0.074946   0.083954  -0.893  0.37201
## contact_telephone             -0.284626   0.040845  -6.968 3.20e-12 ***
## month_may                     -0.636508   0.047352 -13.442  < 2e-16 ***
## month_jun                      0.227993   0.057891   3.938 8.21e-05 ***
## month_jul                      0.352174   0.059033   5.966 2.44e-09 ***
## month_aug                     -0.126775   0.068622  -1.847  0.06468 .
## month_oct                      0.658906   0.089907   7.329 2.32e-13 ***
## month_nov                     -0.180760   0.062345  -2.899  0.00374 **
## month_dec                      0.374553   0.156150   2.399  0.01645 *
```

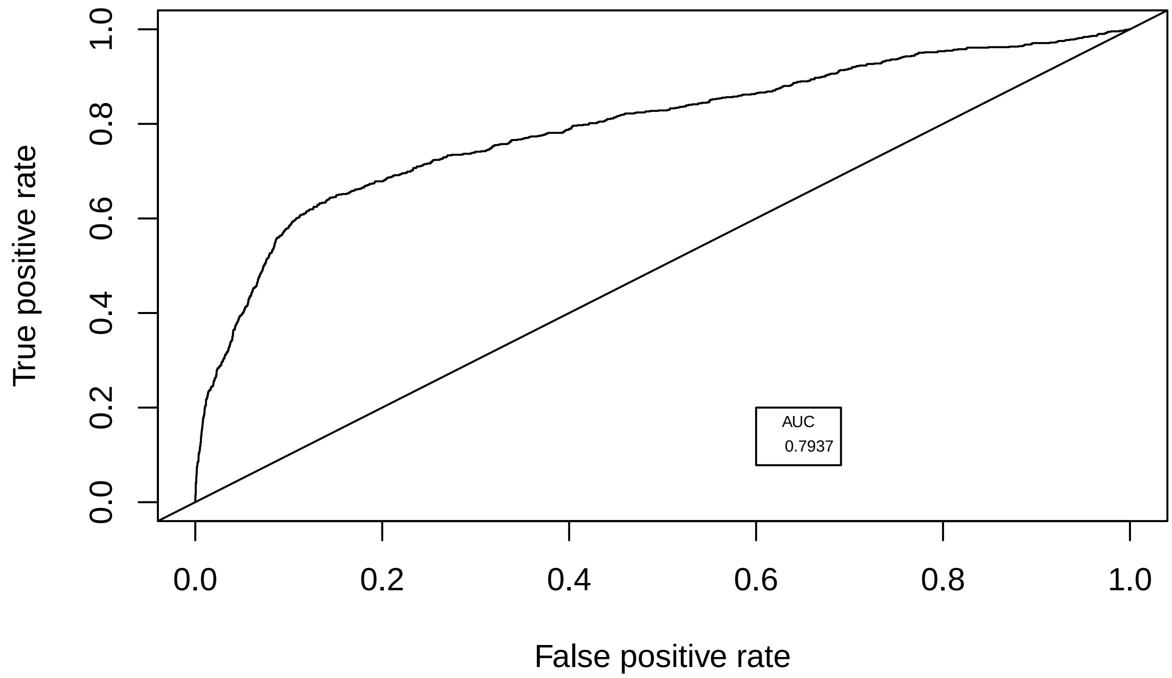
```
## month_mar                     1.058901   0.089983  11.768  < 2e-16 ***
## month_sep                     0.079527   0.095701   0.831  0.40598    
## day_of_week_mon              -0.094884   0.040905  -2.320  0.02036 *  
## day_of_week_tue               0.162650   0.040493   4.017 5.90e-05 ***
## day_of_week_wed               0.128098   0.040952   3.128  0.00176 ** 
## day_of_week_thu               0.069775   0.040013   1.744  0.08119 .  
## poutcome_nonexistent          0.774875   0.070043  11.063  < 2e-16 ***
## poutcome_success              1.860016   0.068073  27.324  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51007  on 41749  degrees of freedom
## Residual deviance: 39812  on 41703  degrees of freedom
## AIC: 39906
##
## Number of Fisher Scoring iterations: 4
```

```r
confusionMatrix(as.factor(ifelse(predict(o_model_lr, type="response",newdata=test) > 0.5,"1","0")), as.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6754  457
##          1  552  470
##
##                Accuracy : 0.8774
##                  95% CI : (0.8702, 0.8845)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 0.997779
##
##                   Kappa : 0.413
##  Mcnemar's Test P-Value : 0.003084
##
##             Sensitivity : 0.50701
##             Specificity : 0.92445
##          Pos Pred Value : 0.45988
##          Neg Pred Value : 0.93662
##              Prevalence : 0.11260
##          Detection Rate : 0.05709
##    Detection Prevalence : 0.12413
##       Balanced Accuracy : 0.71573
##
##        'Positive' Class : 1
##
```

```r
o_pred_lr = predict(o_model_lr, type='response', newdata=test)


pred_lr_o<-prediction(o_pred_lr, test$y)
o_eval_lr <- performance(pred_lr_o,"tpr","fpr")
plot(o_eval_lr, colorize=F)
abline(a=0, b=1)
```

```
o_auc_lr <- performance(pred_lr_o,"auc")
o_auc_lr <- unlist(slot(o_auc_lr,"y.values"))
o_auc_lr <- round(o_auc_lr,4)
legend(.6,.2,o_auc_lr, title="AUC", cex=0.5)
```



#LOGISTIC REGRESSION - UNDER SAMPLING

```
u_model_lr <- glm(formula = y ~ ., data=under, family=binomial)
summary(u_model_lr)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = under)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7053  -0.6787  -0.5241   0.6199   2.5793
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.215060   0.355608   3.417 0.000634 ***
## campaign                      -0.041909   0.011429  -3.667 0.000246 ***
## previous                       0.225355   0.085753   2.628 0.008590 **
## cons.conf.idx                  0.031113   0.006183   5.032 4.85e-07 ***
## euribor3m                     -0.434194   0.019464 -22.307  < 2e-16 ***
## age_1                         -0.186311   0.166544  -1.119 0.263272
## age_2                         -0.335790   0.157059  -2.138 0.032518 *
## age_3                         -0.245169   0.150989  -1.624 0.104429
## job_housemaid                 -0.076696   0.167355  -0.458 0.646750
## job_services                  -0.107228   0.094726  -1.132 0.257643
## job_blue_collar               -0.082917   0.087279  -0.950 0.342101
## job_technician                -0.031789   0.080855  -0.393 0.694201
## job_retired                    0.191268   0.137823   1.388 0.165205
## job_management                -0.117454   0.099040  -1.186 0.235653
## job_unemployed                 0.031079   0.146359   0.212 0.831834
## job_self_employed              0.102092   0.133991   0.762 0.446102
## job_unknown                    0.212257   0.282690   0.751 0.452746
## job_entrepreneur              -0.088654   0.135129  -0.656 0.511777
## job_student                    0.217384   0.141417   1.537 0.124248
## marital_married                0.110061   0.077779   1.415 0.157055
## marital_single                 0.166070   0.088187   1.883 0.059679 .
## marital_unknown                0.630907   0.458292   1.377 0.168621
## education_high.school          0.151551   0.104245   1.454 0.146001
## education_basic.6y             0.173632   0.132497   1.310 0.190042
## education_basic.9y             0.069209   0.105394   0.657 0.511393
## education_professional.course  0.140171   0.116038   1.208 0.227057
## education_unknown              0.291253   0.138737   2.099 0.035789 *
## education_university.degree    0.255207   0.105573   2.417 0.015634 *
## default_unknown               -0.308992   0.071222  -4.338 1.43e-05 ***
## housing_yes                   -0.047104   0.046834  -1.006 0.314532
## housing_unknown               -0.095479   0.155019  -0.616 0.537948
## contact_telephone             -0.223880   0.075613  -2.961 0.003068 **
## month_may                     -0.550543   0.085907  -6.409 1.47e-10 ***
## month_jun                      0.253774   0.104455   2.430 0.015119 *
## month_jul                      0.336105   0.107518   3.126 0.001772 **
## month_aug                     -0.052676   0.124681  -0.422 0.672672
## month_oct                      0.555569   0.159864   3.475 0.000510 ***
## month_nov                     -0.106650   0.113258  -0.942 0.346369
## month_dec                      0.665535   0.280641   2.371 0.017717 *
## month_mar                      1.376902   0.179502   7.671 1.71e-14 ***
```

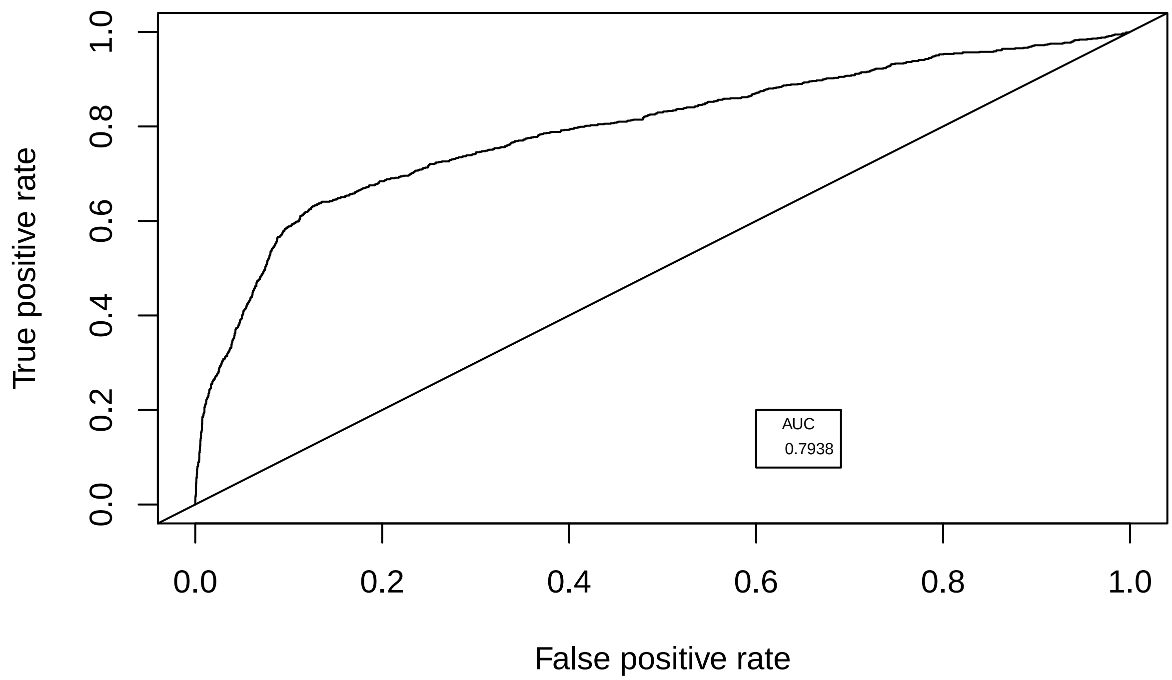
```
## month_sep                          0.172012   0.174701   0.985 0.324815
## day_of_week_mon                   -0.134537   0.074711  -1.801 0.071739 .
## day_of_week_tue                    0.145380   0.074682   1.947 0.051578 .
## day_of_week_wed                    0.149088   0.074453   2.002 0.045238 *
## day_of_week_thu                    0.097529   0.072648   1.343 0.179434
## poutcome_nonexistent               0.669635   0.128199   5.223 1.76e-07 ***
## poutcome_success                   1.752820   0.121763  14.395  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 15104  on 12362  degrees of freedom
## Residual deviance: 11840  on 12316  degrees of freedom
## AIC: 11934
##
## Number of Fisher Scoring iterations: 4
```

```
confusionMatrix(as.factor(ifelse(predict(u_model_lr, type="response",newdata=test) > 0.5,"1","0")), as.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6745  453
##          1  561  474
##
##                Accuracy : 0.8768
##                  95% CI : (0.8695, 0.8839)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 0.9987147
##
##                   Kappa : 0.4135
##
##  Mcnemar's Test P-Value : 0.0007789
##
##             Sensitivity : 0.51133
##             Specificity : 0.92321
##          Pos Pred Value : 0.45797
##          Neg Pred Value : 0.93707
##              Prevalence : 0.11260
##          Detection Rate : 0.05757
##    Detection Prevalence : 0.12571
##       Balanced Accuracy : 0.71727
##
##        'Positive' Class : 1
##
```

```
u_pred_lr = predict(u_model_lr, type='response', newdata=test[-1])

pred_lr_u<-prediction(u_pred_lr, test$y)
u_eval_lr <- performance(pred_lr_u,"tpr","fpr")
plot(u_eval_lr, colorize=F)
abline(a=0, b=1)

u_auc_lr <- performance(pred_lr_u,"auc")
```

```r
u_auc_lr <- unlist(slot(u_auc_lr,"y.values"))
u_auc_lr <- round(u_auc_lr,4)
legend(.6,.2,u_auc_lr, title="AUC", cex=0.5)
```



#LOGISTIC REGRESSION - COMBINATION SAMPLING

```r
b_model_lr <- glm(formula = y ~ ., data=both, family=binomial)
summary(b_model_lr)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = both)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0115  -0.8817  -0.4440   0.8507   2.2703
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    2.767159   0.222832  12.418  < 2e-16 ***
## campaign                      -0.049096   0.006151  -7.981 1.45e-15 ***
## previous                       0.285651   0.055673   5.131 2.88e-07 ***
## cons.conf.idx                  0.043593   0.003881  11.232  < 2e-16 ***
## euribor3m                     -0.428481   0.011628 -36.850  < 2e-16 ***
## age_1                         -0.395382   0.106497  -3.713 0.000205 ***
## age_2                         -0.543282   0.101932  -5.330 9.83e-08 ***
## age_3                         -0.472703   0.098434  -4.802 1.57e-06 ***
## job_housemaid                 -0.197054   0.098428  -2.002 0.045282 *
## job_services                   0.042076   0.052381   0.803 0.421825
## job_blue_collar               -0.071460   0.048809  -1.464 0.143175
## job_technician                -0.047184   0.045759  -1.031 0.302474
## job_retired                    0.129592   0.082483   1.571 0.116152
## job_management                -0.133369   0.056286  -2.369 0.017813 *
## job_unemployed                 0.057176   0.089277   0.640 0.521890
## job_self_employed             -0.004026   0.074048  -0.054 0.956644
## job_unknown                    0.290989   0.160850   1.809 0.070439 .
## job_entrepreneur               0.035331   0.075715   0.467 0.640766
## job_student                    0.444325   0.088250   5.035 4.78e-07 ***
## marital_married                0.058757   0.044006   1.335 0.181808
## marital_single                 0.100890   0.049878   2.023 0.043100 *
## marital_unknown                0.864593   0.272721   3.170 0.001523 **
## education_high.school          0.121146   0.059049   2.052 0.040208 *
## education_basic.6y             0.228986   0.072647   3.152 0.001621 **
## education_basic.9y             0.142938   0.058222   2.455 0.014087 *
## education_professional.course  0.170750   0.065490   2.607 0.009127 **
## education_unknown              0.153197   0.080151   1.911 0.055960 .
## education_university.degree    0.266076   0.059940   4.439 9.04e-06 ***
## default_unknown               -0.204247   0.037542  -5.440 5.32e-08 ***
## housing_yes                   -0.042223   0.026489  -1.594 0.110939
## housing_unknown               -0.242307   0.087190  -2.779 0.005451 **
## contact_telephone             -0.336919   0.044987  -7.489 6.93e-14 ***
## month_may                     -0.601738   0.050362 -11.948  < 2e-16 ***
## month_jun                      0.279490   0.060925   4.587 4.49e-06 ***
## month_jul                      0.290290   0.063568   4.567 4.96e-06 ***
## month_aug                     -0.203330   0.075119  -2.707 0.006794 **
## month_oct                      0.956523   0.106354   8.994  < 2e-16 ***
## month_nov                     -0.300308   0.066566  -4.511 6.44e-06 ***
## month_dec                      0.615045   0.196369   3.132 0.001736 **
## month_mar                      1.081006   0.104969  10.298  < 2e-16 ***
```

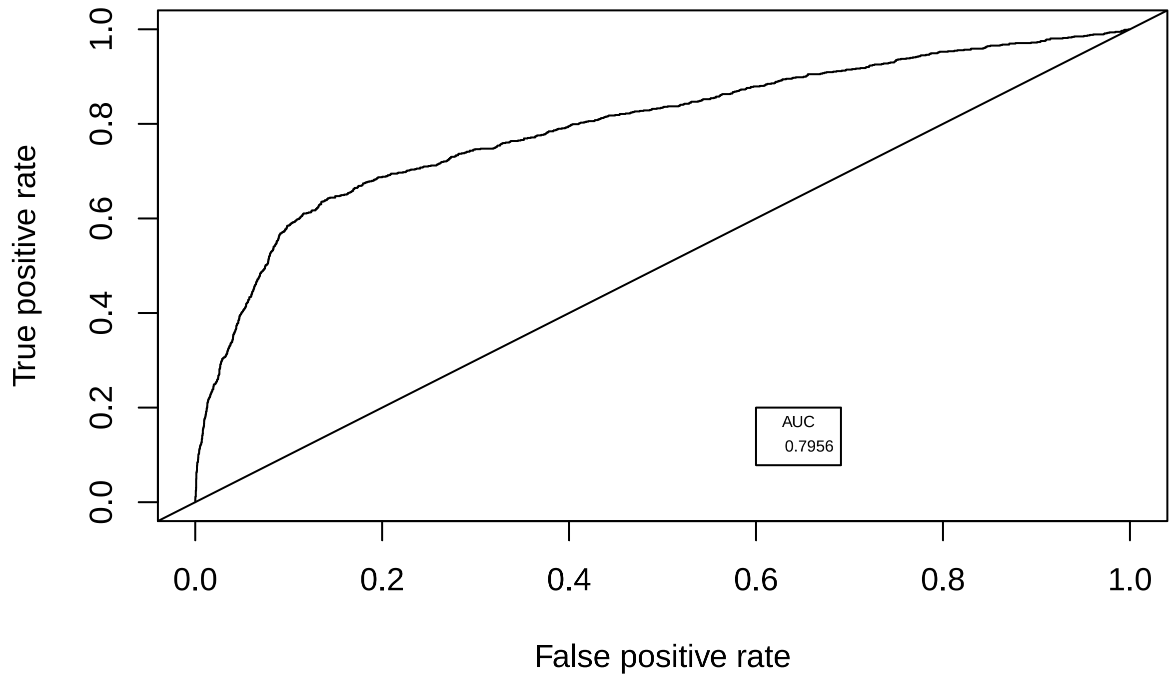
```
## month_sep                        0.173355   0.113489   1.528 0.126635
## day_of_week_mon                 -0.152217   0.041839  -3.638 0.000275 ***
## day_of_week_tue                  0.042873   0.042148   1.017 0.309066
## day_of_week_wed                  0.117725   0.041791   2.817 0.004848 **
## day_of_week_thu                  0.066906   0.041166   1.625 0.104102
## poutcome_nonexistent             0.817976   0.079349  10.309  < 2e-16 ***
## poutcome_success                 1.856392   0.081154  22.875  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 45656  on 32933  degrees of freedom
## Residual deviance: 35431  on 32887  degrees of freedom
## AIC: 35525
##
## Number of Fisher Scoring iterations: 5
```

```
confusionMatrix(as.factor(ifelse(predict(b_model_lr, type="response",newdata=test) > 0.5,"1","0")), as.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6086  319
##          1 1220  608
##
##                Accuracy : 0.8131
##                  95% CI : (0.8045, 0.8214)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3432
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.65588
##             Specificity : 0.83301
##          Pos Pred Value : 0.33260
##          Neg Pred Value : 0.95020
##              Prevalence : 0.11260
##          Detection Rate : 0.07385
##    Detection Prevalence : 0.22203
##       Balanced Accuracy : 0.74445
##
##        'Positive' Class : 1
##
```

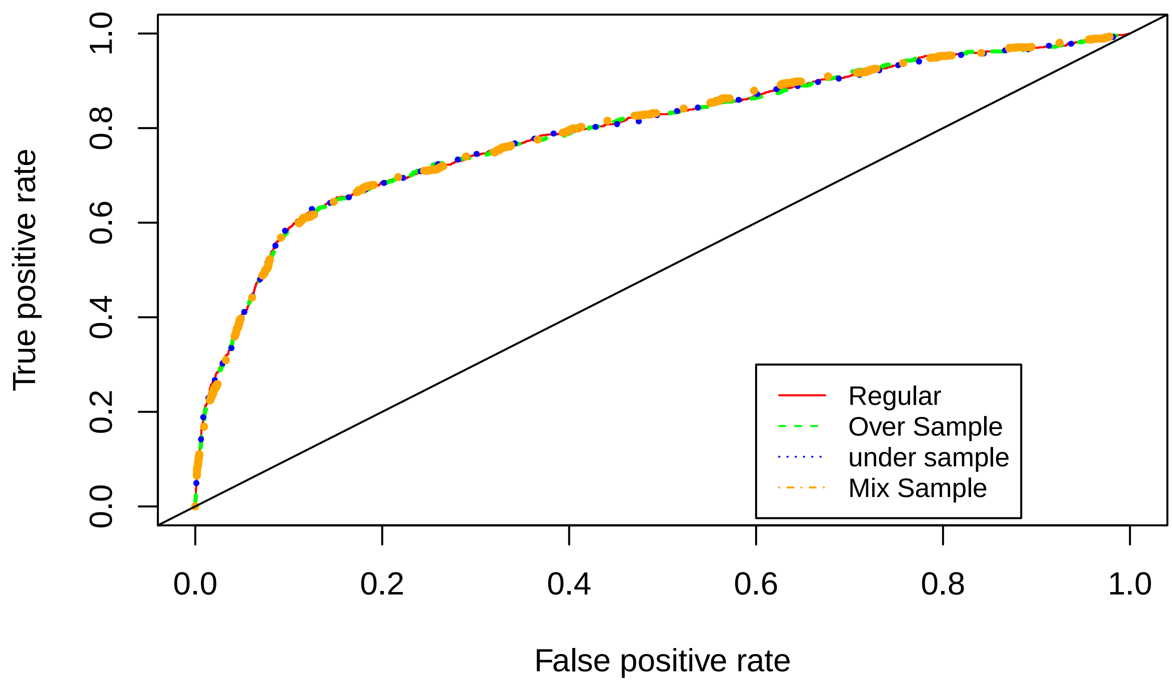
```
b_pred_lr = predict(b_model_lr, type='response', newdata=test[-1])

#Accuracy is calculated at 89%

pred_lr_b<-prediction(b_pred_lr, test$y)
b_eval_lr <- performance(pred_lr_b,"tpr","fpr")
plot(b_eval_lr, colorize=F)
abline(a=0, b=1)
```

```
b_auc_lr <- performance(pred_lr_b,"auc")
b_auc_lr <- unlist(slot(b_auc_lr,"y.values"))
b_auc_lr <- round(b_auc_lr,4)
legend(.6,.2,b_auc_lr, title="AUC", cex=0.5)
```



#LOGISTIC REGRESSION - ROC CURVE COMPARISION

```
plot(r_eval_lr, lty= 1, lwd=1, col = "red", colorize=F)
plot(o_eval_lr, lty=2, lwd= 2, col="green", add=TRUE)
plot(u_eval_lr, lty=3, lwd= 3, col="blue", add=TRUE)
plot(b_eval_lr, lty=4, lwd= 4, col="orange", add=TRUE)
abline(a=0, b=1)
legend(.6, .3, legend=c("Regular", "Over Sample","under sample", "Mix Sample"),
       col=c("red","green" ,"blue","orange"), lty=1:4, cex=0.8)
```



#LOGISTIC REGRESSION - MODEL EVALUATION USING THE K-FOLD CROSS VALIDATION METHOD.

```
folds = createFolds(both$y, k=10)
cv = lapply(folds, function(x){
  train_fold= both[-x,]
  test_fold = test[x,]
  kf_model_lr <- glm(formula = y ~ ., data=train_fold, family=binomial())
  kf_pred_lr = predict(kf_model_lr,  type="response", newdata=test_fold)
  lr_cm = ifelse(kf_pred_lr >= 0.5, 1,0)
  lr_cm_tab = table(lr_cm,test_fold$y)
  accuracy=(lr_cm_tab[1,1]+lr_cm_tab[2,2])/(lr_cm_tab[1,1]+lr_cm_tab[1,2]+lr_cm_tab[2,1]+lr_cm_tab[2,2])
  sensitivity=lr_cm_tab[2,2]/(lr_cm_tab[2,2] + lr_cm_tab[1,2])
  specificity=lr_cm_tab[1,1]/(lr_cm_tab[1,1] + lr_cm_tab[2,1])
  return(data.frame(accuracy, sensitivity, specificity))
})

cv
```

```
## $Fold01
##    accuracy sensitivity specificity
## 1 0.8147296   0.7241379   0.8248082
##
## $Fold02
##    accuracy sensitivity specificity
## 1 0.8299156   0.7326733   0.8434066
##
## $Fold03
##   accuracy sensitivity specificity
## 1 0.817296   0.5963303   0.8511236
##
## $Fold04
##    accuracy sensitivity specificity
## 1 0.8131313   0.6222222   0.8376068
##
## $Fold05
##    accuracy sensitivity specificity
## 1 0.7931873   0.5909091   0.8174387
##
## $Fold06
##    accuracy sensitivity specificity
## 1 0.8221681   0.6263736   0.8465753
##
## $Fold07
##    accuracy sensitivity specificity
## 1 0.7839136   0.6263736   0.8032345
##
## $Fold08
##    accuracy sensitivity specificity
## 1 0.8378063   0.7281553   0.8527851
##
## $Fold09
##    accuracy sensitivity specificity
## 1 0.7985075   0.6741573    0.813986
##
```

```
## $Fold10
##    accuracy sensitivity specificity
## 1 0.8280255   0.6794872    0.844413
```

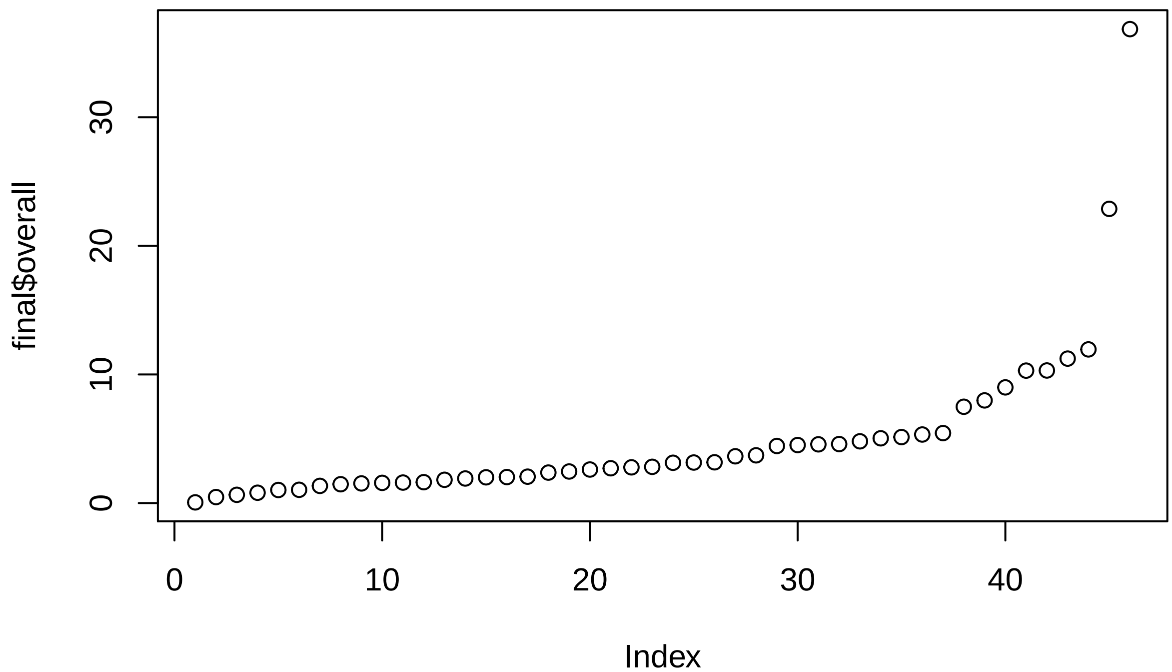
LOGISTIC REGRESSION - VARIABLE IMPORTANCE

```
imp <- as.data.frame(varImp(b_model_lr))
imp <- data.frame(overall = imp$Overall,
                  names   = rownames(imp))
final<-imp[order(imp$overall,decreasing = F),]
final
```

```
##        overall                         names
## 15  0.05436488             job_self_employed
## 17  0.46662793              job_entrepreneur
## 14  0.64043426                job_unemployed
## 9   0.80325938                  job_services
## 42  1.01718451               day_of_week_tue
## 11  1.03114172                job_technician
## 19  1.33520829               marital_married
## 10  1.46407051               job_blue_collar
## 40  1.52750666                     month_sep
## 12  1.57113127                   job_retired
## 29  1.59398883                   housing_yes
## 44  1.62528614               day_of_week_thu
## 16  1.80907783                   job_unknown
## 26  1.91135049             education_unknown
## 8   2.00202181                 job_housemaid
## 20  2.02273975                marital_single
## 22  2.05160056         education_high.school
## 13  2.36947710                job_management
## 24  2.45502540            education_basic.9y
## 25  2.60725332 education_professional.course
## 35  2.70677272                     month_aug
## 30  2.77908466               housing_unknown
## 43  2.81697487               day_of_week_wed
## 38  3.13208672                     month_dec
## 23  3.15204851            education_basic.6y
## 21  3.17025125               marital_unknown
## 41  3.63814044               day_of_week_mon
## 5   3.71262735                         age_1
## 27  4.43904064   education_university.degree
## 37  4.51145281                     month_nov
## 34  4.56657695                     month_jul
## 33  4.58742983                     month_jun
## 7   4.80222791                         age_3
## 18  5.03484627                   job_student
## 2   5.13087037                      previous
## 6   5.32982354                         age_2
## 28  5.44043555               default_unknown
## 31  7.48918927             contact_telephone
## 1   7.98123570                      campaign
## 36  8.99378225                     month_oct
## 39 10.29829996                     month_mar
## 45 10.30852618           poutcome_nonexistent
## 3  11.23234363                 cons.conf.idx
## 32 11.94829427                     month_may
## 46 22.87483515              poutcome_success
```

```
## 4  36.85048468                          euribor3m
```

```
plot(final$overall)
```



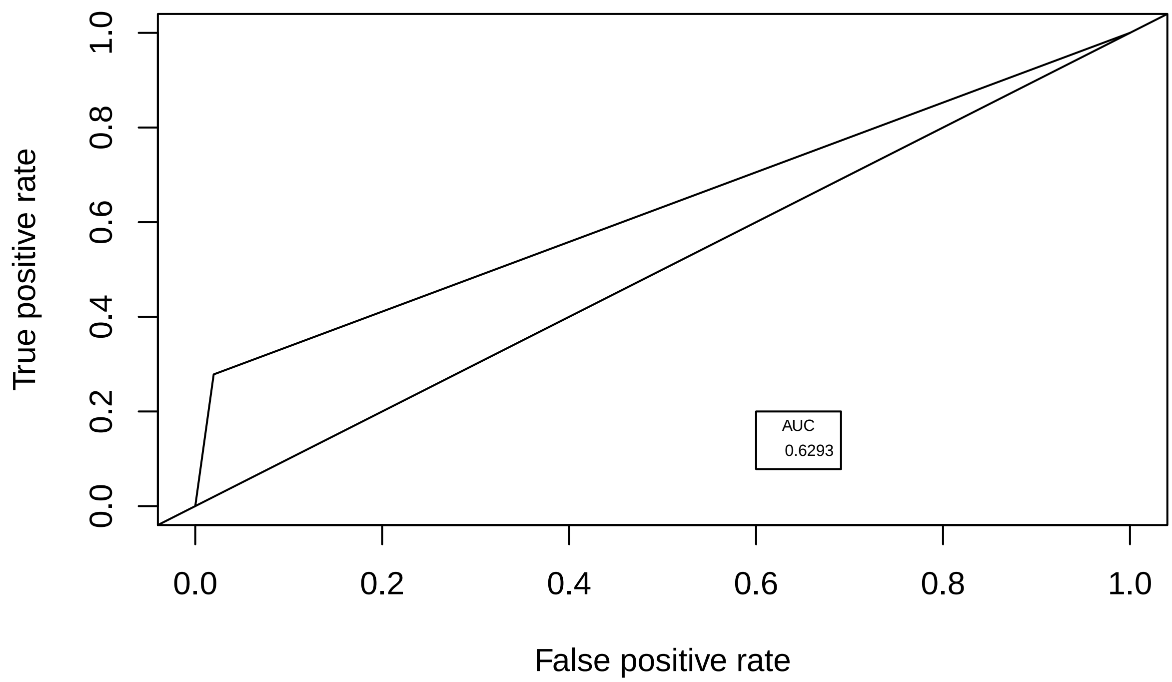
#RANDOM FOREST

FITTING RANDOM FOREST MODEL TO THE TRAINING DATASET

```r
r_model_rf <- randomForest(factor(y) ~.,data=train, ntrees = 500)

confusionMatrix(predict(r_model_rf, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7162  668
##          1  144  259
##
##                Accuracy : 0.9014
##                  95% CI : (0.8947, 0.9077)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 2.418e-05
##
##                   Kappa : 0.3448
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.27940
##             Specificity : 0.98029
##          Pos Pred Value : 0.64268
##          Neg Pred Value : 0.91469
##              Prevalence : 0.11260
##          Detection Rate : 0.03146
##    Detection Prevalence : 0.04895
##       Balanced Accuracy : 0.62984
##
##        'Positive' Class : 1
##
```

```r
r_pred_rf = predict(r_model_rf,  type="class", newdata=test)

pred_rf_r<-prediction(as.numeric(r_pred_rf), test$y)

r_eval_rf <- performance(pred_rf_r,"tpr","fpr")
plot(r_eval_rf, colorize=F)
abline(a=0, b=1)

r_auc_rf <- performance(pred_rf_r,"auc")
r_auc_rf <- unlist(slot(r_auc_rf,"y.values"))
r_auc_rf <- round(r_auc_rf,4)
legend(.6,.2,r_auc_rf, title="AUC", cex=0.5)
```



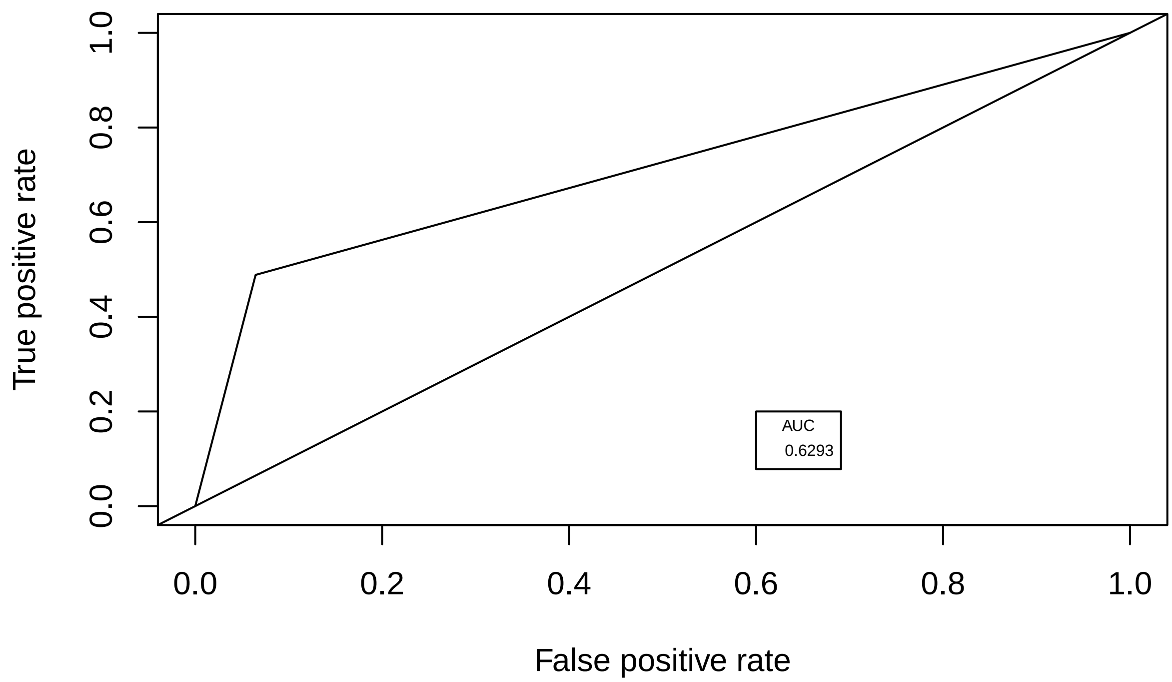
1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
AUC
0.6293
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

#RANDOM FOREST - OVER SAMPLING

```
o_model_rf <- randomForest(factor(y) ~.,data=over, ntrees = 500)

confusionMatrix(predict(o_model_rf, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6835  474
##          1  471  453
##
##                Accuracy : 0.8852
##                  95% CI : (0.8781, 0.892)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 0.7414
##
##                   Kappa : 0.4248
##  Mcnemar's Test P-Value : 0.9481
##
##             Sensitivity : 0.48867
##             Specificity : 0.93553
##          Pos Pred Value : 0.49026
##          Neg Pred Value : 0.93515
##              Prevalence : 0.11260
##          Detection Rate : 0.05502
##    Detection Prevalence : 0.11223
##       Balanced Accuracy : 0.71210
##
##        'Positive' Class : 1
##
```

```
o_pred_rf = predict(o_model_rf,  type="class", newdata=test)

pred_rf_o<-prediction(as.numeric(o_pred_rf), test$y)

o_eval_rf <- performance(pred_rf_o,"tpr","fpr")
plot(o_eval_rf, colorize=F)
abline(a=0, b=1)

o_auc_rf <- performance(pred_rf_o,"auc")
o_auc_rf <- unlist(slot(o_auc_rf,"y.values"))
o_auc_rf <- round(o_auc_rf,4)
legend(.6,.2,r_auc_rf, title="AUC", cex=0.5)
```

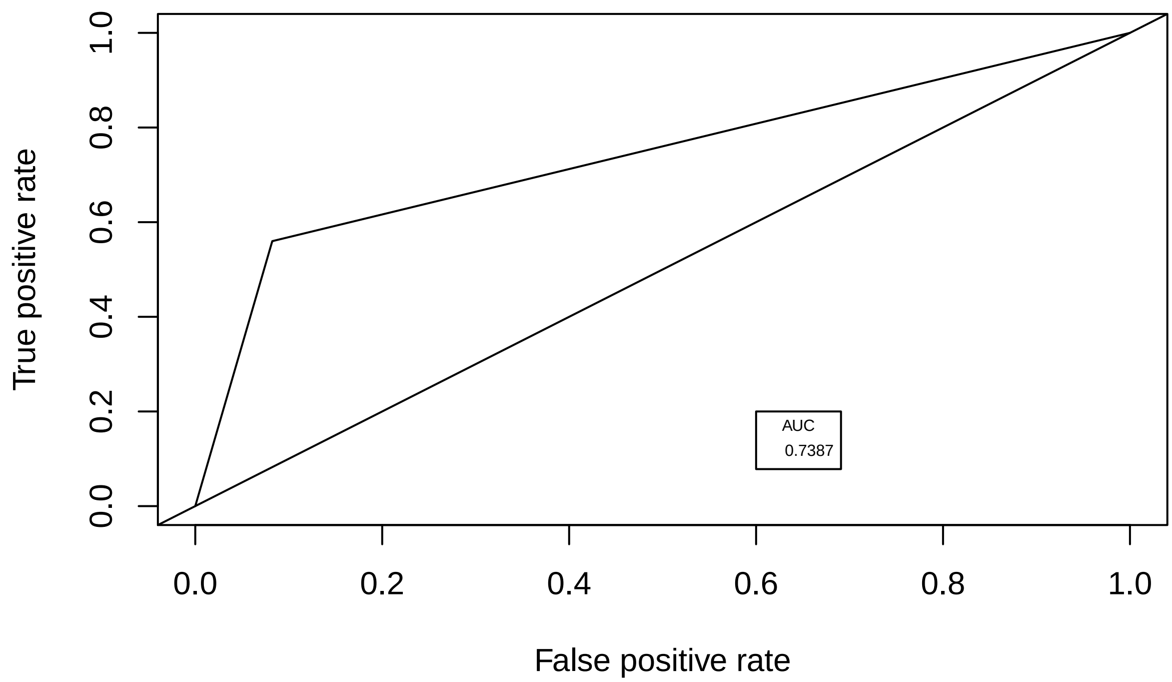

True positive rate
1.0
0.8
0.6
0.4
0.2
0.0
AUC
0.6293
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

#RANDOM FOREST - UNDER SAMPLING

```
u_model_rf <- randomForest(factor(y) ~.,data=under, ntrees = 500)

confusionMatrix(predict(u_model_rf, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6703  409
##          1  603  518
##
##                Accuracy : 0.8771
##                  95% CI : (0.8698, 0.8841)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 0.9984
##
##                   Kappa : 0.4364
##  Mcnemar's Test P-Value : 1.304e-09
##
##             Sensitivity : 0.55879
##             Specificity : 0.91747
##          Pos Pred Value : 0.46209
##          Neg Pred Value : 0.94249
##              Prevalence : 0.11260
##          Detection Rate : 0.06292
##    Detection Prevalence : 0.13616
##       Balanced Accuracy : 0.73813
##
##        'Positive' Class : 1
##
```

```
u_pred_rf = predict(u_model_rf,  type="class", newdata=test)

pred_rf_u<-prediction(as.numeric(u_pred_rf), test$y)

u_eval_rf <- performance(pred_rf_u,"tpr","fpr")
plot(u_eval_rf, colorize=F)
abline(a=0, b=1)


u_auc_rf <- performance(pred_rf_u,"auc")
u_auc_rf <- unlist(slot(u_auc_rf,"y.values"))
u_auc_rf <- round(u_auc_rf,4)
legend(.6,.2,u_auc_rf, title="AUC", cex=0.5)
```



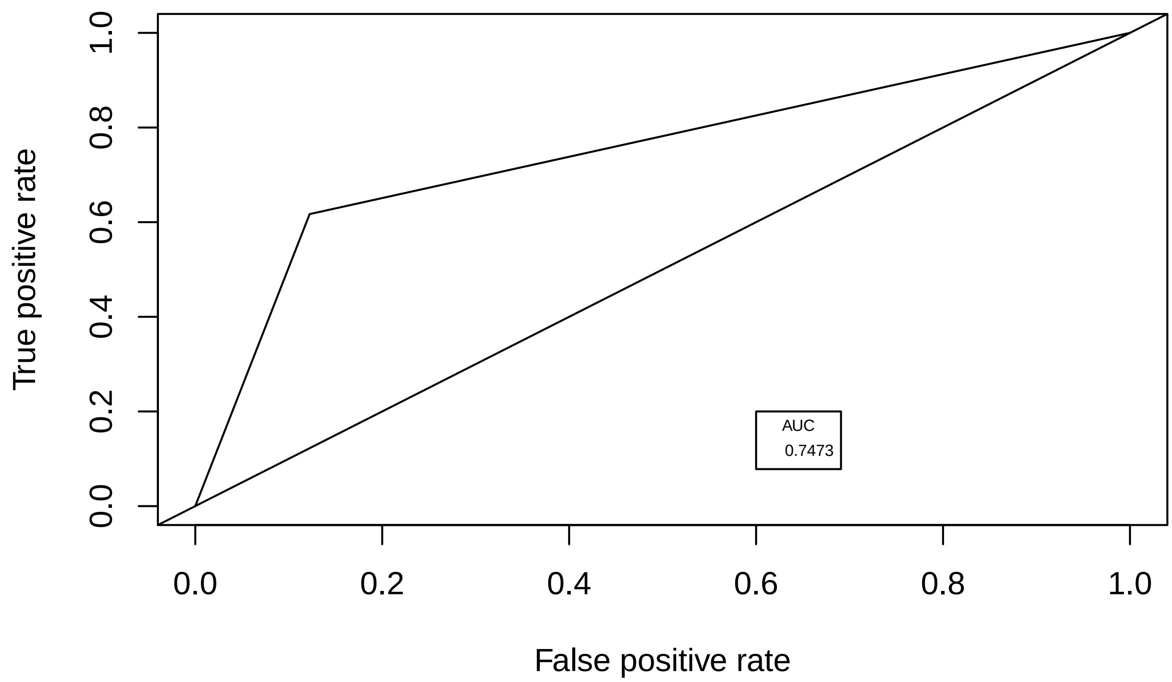
True positive rate
1.0
0.8
0.6
0.4
0.2
0.0
AUC
0.7387
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

#RANDOM FOREST - COMBINATION SAMPLING

```
b_model_rf <- randomForest(factor(y) ~.,data=both, ntrees = 500)

confusionMatrix(predict(b_model_rf, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6413  356
##          1  893  571
##
##                Accuracy : 0.8483
##                  95% CI : (0.8404, 0.856)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3941
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.61597
##             Specificity : 0.87777
##          Pos Pred Value : 0.39003
##          Neg Pred Value : 0.94741
##              Prevalence : 0.11260
##          Detection Rate : 0.06936
##    Detection Prevalence : 0.17782
##       Balanced Accuracy : 0.74687
##
##        'Positive' Class : 1
##
```

```
b_pred_rf = predict(b_model_rf,  type="class", newdata=test)

pred_rf_b<-prediction(as.numeric(b_pred_rf), test$y)

b_eval_rf <- performance(pred_rf_b,"tpr","fpr")
plot(b_eval_rf, colorize=F)
abline(a=0, b=1)


b_auc_rf <- performance(pred_rf_b,"auc")
b_auc_rf <- unlist(slot(b_auc_rf,"y.values"))
b_auc_rf <- round(b_auc_rf,4)
legend(.6,.2,b_auc_rf, title="AUC", cex=0.5)
```

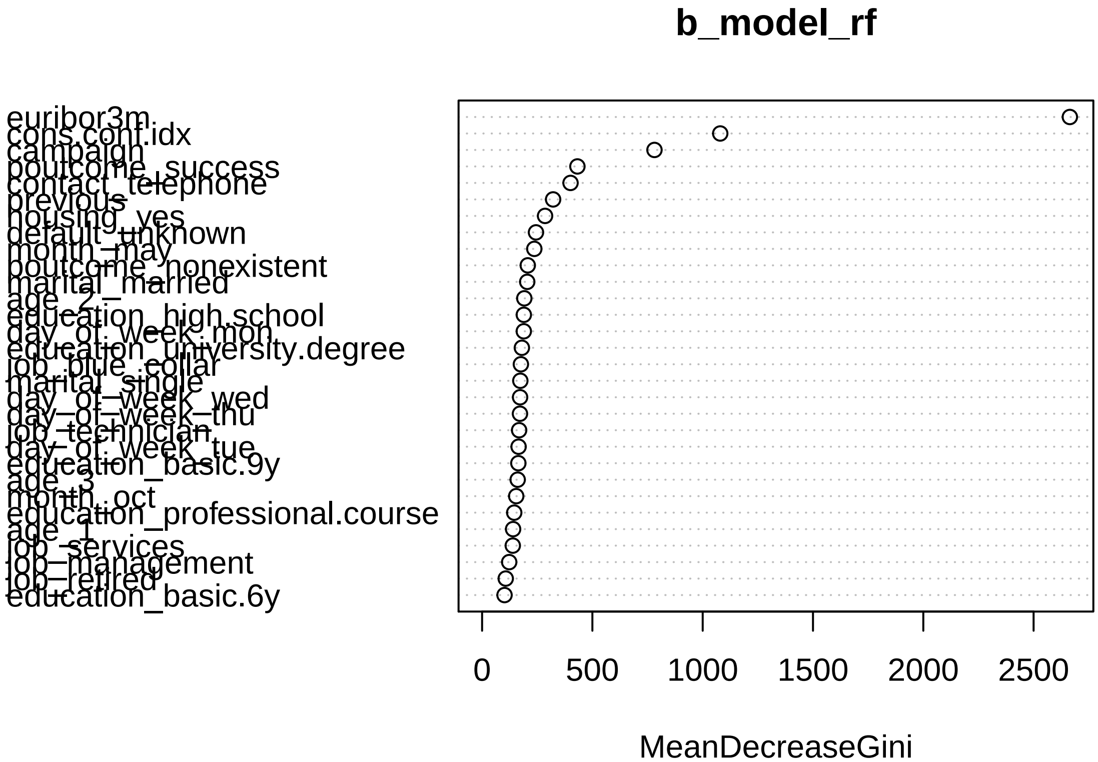

True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC
0.7473
False positive rate

#RANDOM FOREST - VARIABLE IMPORTANCE

```
varImp(b_model_rf)
```

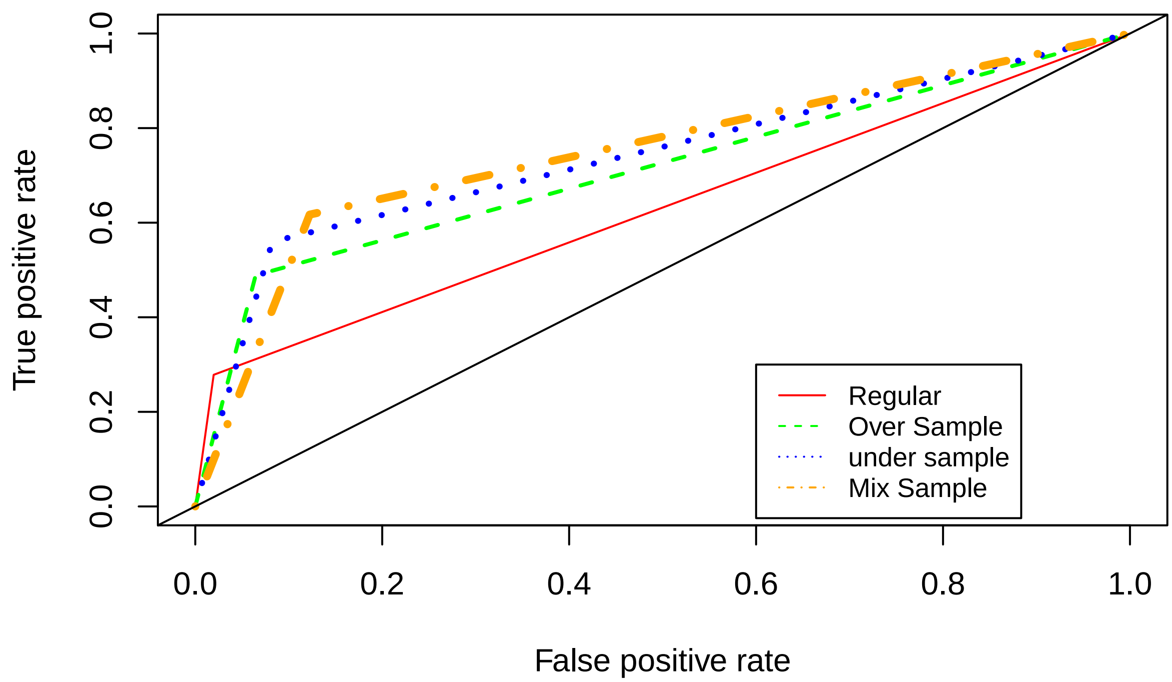
```
##                                 Overall
## campaign                      781.33973
## previous                      321.94606
## cons.conf.idx                1079.52822
## euribor3m                    2664.27691
## age_1                         140.56586
## age_2                         191.39329
## age_3                         161.33619
## job_housemaid                  53.60966
## job_services                  139.16038
## job_blue_collar               175.91056
## job_technician                167.86247
## job_retired                   107.18006
## job_management                122.92106
## job_unemployed                 63.29034
## job_self_employed              89.76119
## job_unknown                    31.95353
## job_entrepreneur               94.65120
## job_student                    79.99798
## marital_married               204.60277
## marital_single                173.36865
## marital_unknown                18.62605
## education_high.school         189.23424
## education_basic.6y            101.69603
## education_basic.9y            164.41646
## education_professional.course  145.70276
## education_unknown              94.90305
## education_university.degree   180.37949
## default_unknown               244.37427
## housing_yes                   285.46593
## housing_unknown                62.73538
## contact_telephone             400.90597
## month_may                     236.74802
## month_jun                      95.56536
## month_jul                      74.97339
## month_aug                      69.83970
## month_oct                     154.76943
## month_nov                      75.91114
## month_dec                      15.99606
## month_mar                      92.38439
## month_sep                      62.53214
## day_of_week_mon               189.01743
## day_of_week_tue               165.44155
## day_of_week_wed               172.19233
## day_of_week_thu               171.73888
## poutcome_nonexistent          207.07789
## poutcome_success              432.12290
```

```
varImpPlot(b_model_rf)
```


b_model_rf
euribor3m
cons.conf.idx
campaign
poutcome_success
contact_telephone
previous
housing_yes
default_unknown
month_may
poutcome_nonexistent
marital_married
age_2
education_high.school
day_of_week_mon
education_university.degree
job_blue.collar
marital_single
day_of_week_wed
day_of_week_thu
job_technician
day_of_week_tue
education_basic.9y
age_3
month_oct
education_professional.course
age_1
job_services
job_management
job_retired
education_basic.6y
0
500
1000
1500
2000
2500
MeanDecreaseGini

#RANDOM FOREST - ROC CURVE COMPARISION

```
plot(r_eval_rf, lty= 1, lwd=1, col = "red", colorize=F)
plot(o_eval_rf, lty=2, lwd= 2, col="green", add=TRUE)
plot(u_eval_rf, lty=3, lwd= 3, col="blue", add=TRUE)
plot(b_eval_rf, lty=4, lwd= 4, col="orange", add=TRUE)
abline(a=0, b=1)
legend(.6, .3, legend=c("Regular", "Over Sample","under sample", "Mix Sample"),
       col=c("red","green" ,"blue","orange"), lty=1:4, cex=0.8)
```



RANDOM FOREST - MODEL EVALUATION USING THE K-FOLD CROSS VALIDATION METHOD.

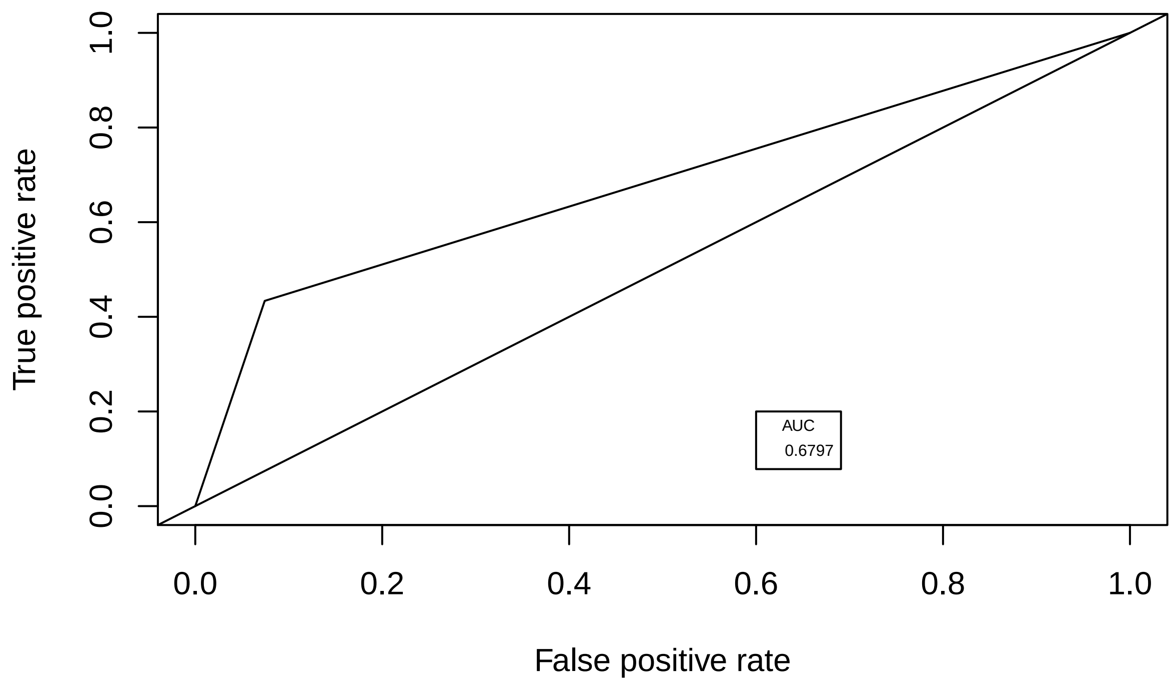
```
folds = createFolds(both$y, k=10)
cv_rf = lapply(folds, function(x){
  train_fold= both[-x,]
  test_fold = test[x,]
  kf_model_rf <- randomForest(factor(y) ~.,data=train_fold, ntrees = 500)
  kf_pred_rf = predict(kf_model_rf,  type="class", newdata=test_fold)
  rf_cm = ifelse(kf_pred_rf == 0, 0, 1)
  rf_cm_tab = table(rf_cm,test_fold$y)
  accuracy=(rf_cm_tab[1,1]+rf_cm_tab[2,2])/(rf_cm_tab[1,1]+rf_cm_tab[1,2]+rf_cm_tab[2,1]+rf_cm_tab[2,2])
  sensitivity=rf_cm_tab[2,2]/(rf_cm_tab[2,2] + rf_cm_tab[1,2])
  specificity=rf_cm_tab[1,1]/(rf_cm_tab[1,1] + rf_cm_tab[2,1])
  return(data.frame(accuracy, sensitivity, specificity))
})

#cv_rf
```

FITTING NAIVE BAYES MODEL

```
r_model_nb = naiveBayes( factor(y) ~., data=train, importance=T)

confusionMatrix(predict(r_model_nb, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6763  525
##          1  543  402
##
##                Accuracy : 0.8703
##                  95% CI : (0.8628, 0.8775)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.3563
##  Mcnemar's Test P-Value : 0.6029
##
##             Sensitivity : 0.43366
##             Specificity : 0.92568
##          Pos Pred Value : 0.42540
##          Neg Pred Value : 0.92796
##              Prevalence : 0.11260
##          Detection Rate : 0.04883
##    Detection Prevalence : 0.11478
##       Balanced Accuracy : 0.67967
##
##        'Positive' Class : 1
##
```

```
r_pred_nb = predict(r_model_nb,  type="class", newdata=test)

pred_nb_r<-prediction(as.numeric(r_pred_nb), test$y)

r_eval_nb <- performance(pred_nb_r,"tpr","fpr")
plot(r_eval_nb, colorize=F)
abline(a=0, b=1)


r_auc_nb <- performance(pred_nb_r,"auc")
r_auc_nb <- unlist(slot(r_auc_nb,"y.values"))
r_auc_nb <- round(r_auc_nb,4)
legend(.6,.2,r_auc_nb, title="AUC", cex=0.5)
```

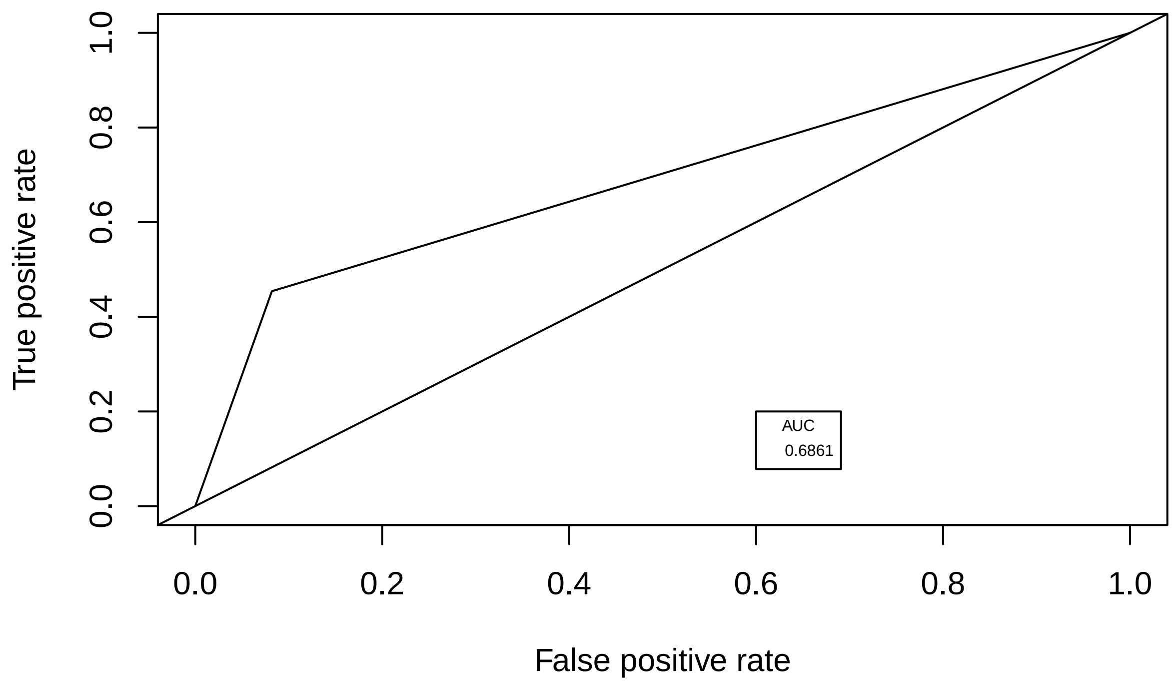

True positive rate
1.0
0.8
0.6
0.4
0.2
0.0
AUC
0.6797
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

#NAIVE BAYES - OVER SAMPLING

```r
o_model_nb = naiveBayes( factor(y) ~., data=over, importance=T)

confusionMatrix(predict(o_model_nb, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6707  506
##          1  599  421
##
##                Accuracy : 0.8658
##                  95% CI : (0.8582, 0.8731)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 1.000000
##
##                   Kappa : 0.3565
##  Mcnemar's Test P-Value : 0.005647
##
##             Sensitivity : 0.45415
##             Specificity : 0.91801
##          Pos Pred Value : 0.41275
##          Neg Pred Value : 0.92985
##              Prevalence : 0.11260
##          Detection Rate : 0.05114
##    Detection Prevalence : 0.12389
##       Balanced Accuracy : 0.68608
##
##        'Positive' Class : 1
##
```

```r
o_pred_nb = predict(o_model_nb,  type="class", newdata=test)

pred_nb_o<-prediction(as.numeric(o_pred_nb), test$y)

o_eval_nb <- performance(pred_nb_o,"tpr","fpr")
plot(o_eval_nb, colorize=F)
abline(a=0, b=1)


o_auc_nb <- performance(pred_nb_o,"auc")
o_auc_nb <- unlist(slot(o_auc_nb,"y.values"))
o_auc_nb <- round(o_auc_nb,4)
legend(.6,.2,o_auc_nb, title="AUC", cex=0.5)
```



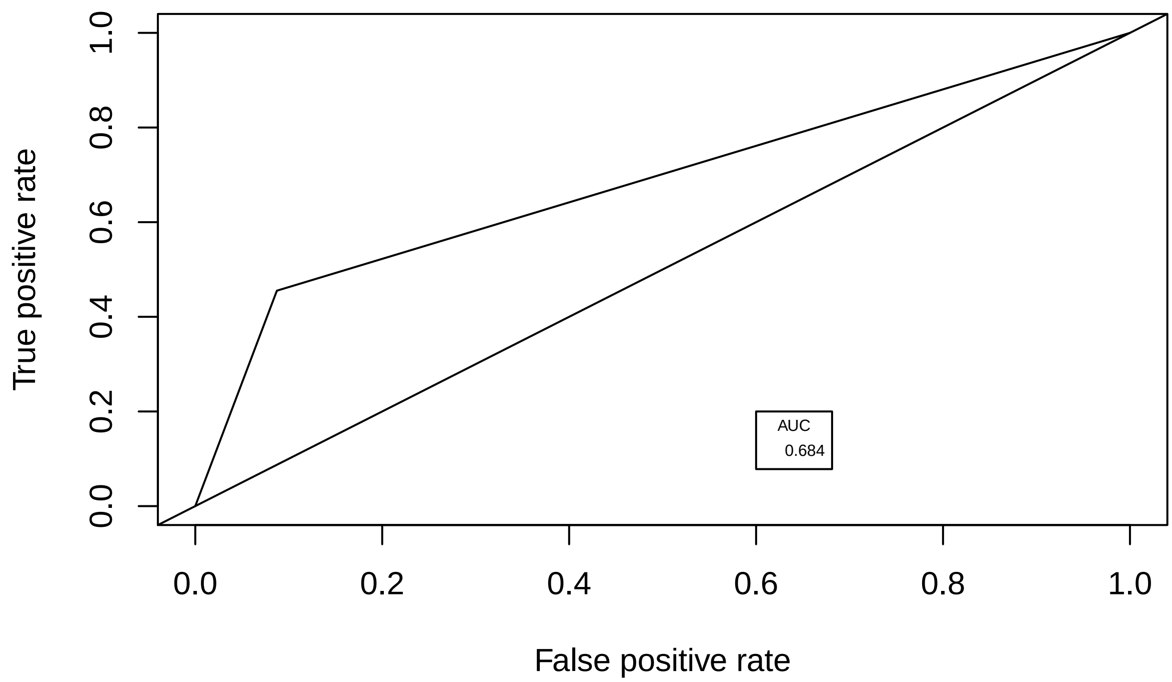
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC
0.6861
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

#NAIVE BAYES - UNDER SAMPLING

```
u_model_nb = naiveBayes( factor(y) ~., data=under, importance=T)

confusionMatrix(predict(u_model_nb, test), as.factor(test$y), positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6668  505
##          1  638  422
##
##                Accuracy : 0.8612
##                  95% CI : (0.8535, 0.8686)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3462
##  Mcnemar's Test P-Value : 9.447e-05
##
##             Sensitivity : 0.45523
##             Specificity : 0.91267
##          Pos Pred Value : 0.39811
##          Neg Pred Value : 0.92960
##              Prevalence : 0.11260
##          Detection Rate : 0.05126
##    Detection Prevalence : 0.12875
##       Balanced Accuracy : 0.68395
##
##        'Positive' Class : 1
##
```

```
u_pred_nb = predict(u_model_nb,  type="class", newdata=test)

pred_nb_u<-prediction(as.numeric(u_pred_nb), test$y)

u_eval_nb <- performance(pred_nb_u,"tpr","fpr")
plot(u_eval_nb, colorize=F)
abline(a=0, b=1)


u_auc_nb <- performance(pred_nb_u,"auc")
u_auc_nb <- unlist(slot(u_auc_nb,"y.values"))
u_auc_nb <- round(u_auc_nb,4)
legend(.6,.2,u_auc_nb, title="AUC", cex=0.5)
```

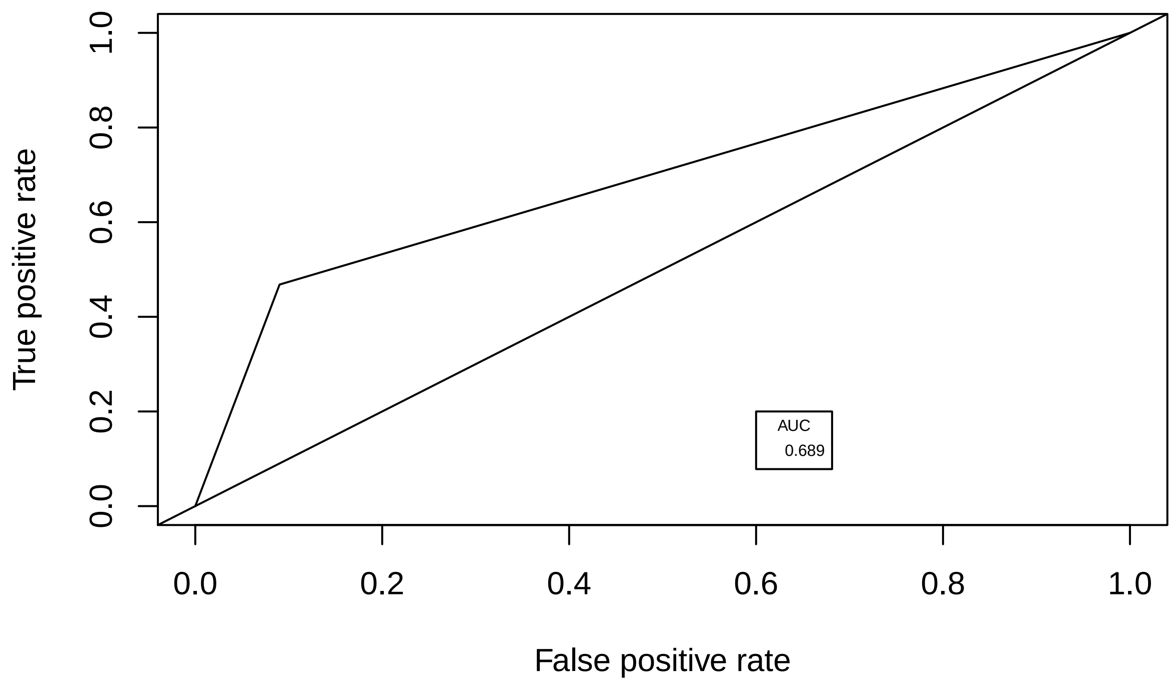

AUC
0.684
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

#NAIVE BAYES - COMBINATION SAMPLING

```
b_model_nb = naiveBayes( factor(y) ~., data=both, importance=T)

confusionMatrix(predict(b_model_nb, test), as.factor(test$y), positive="1")
```

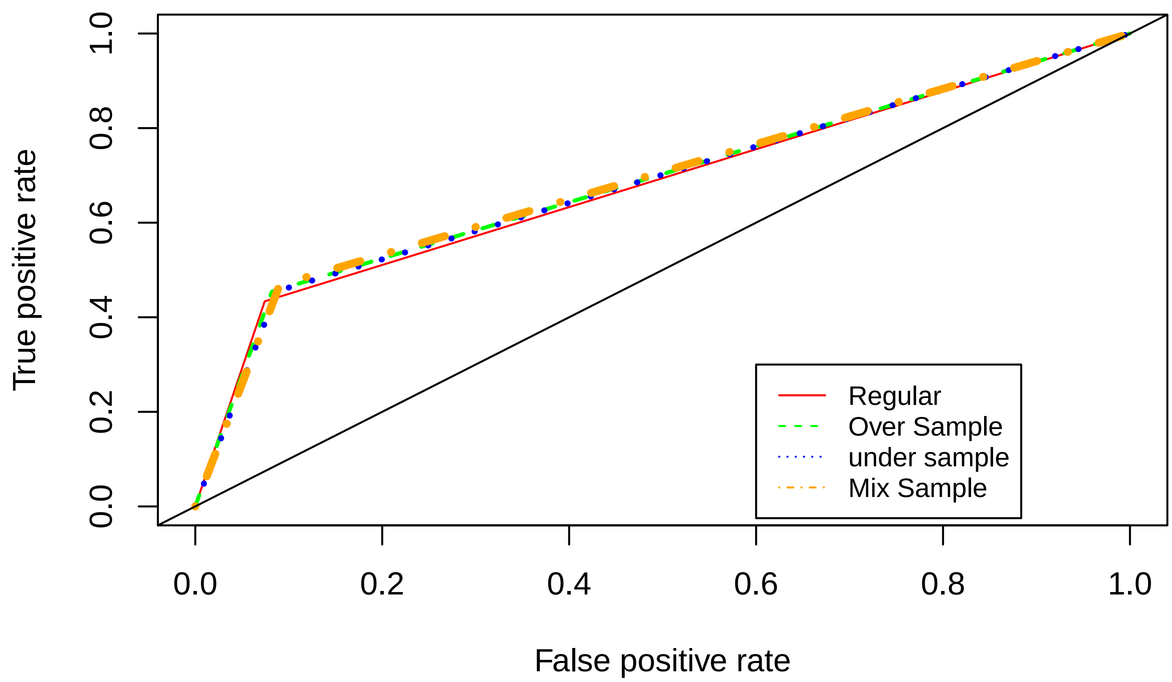
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6647  493
##          1  659  434
##
##                Accuracy : 0.8601
##                  95% CI : (0.8524, 0.8675)
##     No Information Rate : 0.8874
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3506
##  Mcnemar's Test P-Value : 1.166e-06
##
##             Sensitivity : 0.46818
##             Specificity : 0.90980
##          Pos Pred Value : 0.39707
##          Neg Pred Value : 0.93095
##              Prevalence : 0.11260
##          Detection Rate : 0.05271
##    Detection Prevalence : 0.13276
##       Balanced Accuracy : 0.68899
##
##        'Positive' Class : 1
##
```

```
b_pred_nb = predict(b_model_nb,  type="class", newdata=test)

pred_nb_b<-prediction(as.numeric(b_pred_nb), test$y)

b_eval_nb <- performance(pred_nb_b,"tpr","fpr")
plot(b_eval_nb, colorize=F)
abline(a=0, b=1)


b_auc_nb <- performance(pred_nb_b,"auc")
b_auc_nb <- unlist(slot(b_auc_nb,"y.values"))
b_auc_nb <- round(b_auc_nb,4)
legend(.6,.2,b_auc_nb, title="AUC", cex=0.5)
```


1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
AUC
0.689
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate

#NAIVE BAYES - ROC CURVE COMPARISION

```
plot(r_eval_nb, lty= 1, lwd=1, col = "red", colorize=F)
plot(o_eval_nb, lty=2, lwd= 2, col="green", add=TRUE)
plot(u_eval_nb, lty=3, lwd= 3, col="blue", add=TRUE)
plot(b_eval_nb, lty=4, lwd= 4, col="orange", add=TRUE)
abline(a=0, b=1)
legend(.6, .3, legend=c("Regular", "Over Sample","under sample", "Mix Sample"),
       col=c("red","green" ,"blue","orange"), lty=1:4, cex=0.8)
```



NAIVE BAYES - MODEL EVALUATION USING THE K-FOLD CROSS VALIDATION METHOD.

```
folds = createFolds(both$y, k=10)
cv_nb = lapply(folds, function(x){
  train_fold= both[-x,]
  test_fold = test[x,]
  model_nb = naiveBayes( factor(y) ~., data=train_fold, importance=T)
  pred_nb = predict(model_nb,  type="class", newdata=test_fold)
  nb_cm = ifelse(pred_nb == 1, 1,0)
  nb_cm_tab = table(nb_cm,test_fold$y)
  accuracy=(nb_cm_tab[1,1]+nb_cm_tab[2,2])/(nb_cm_tab[1,1]+nb_cm_tab[1,2]+nb_cm_tab[2,1]+nb_cm_tab[2,2])
  sensitivity=nb_cm_tab[2,2]/(nb_cm_tab[2,2] + nb_cm_tab[1,2])
  specificity=nb_cm_tab[1,1]/(nb_cm_tab[1,1] + nb_cm_tab[2,1])
  return(data.frame(accuracy, sensitivity, specificity))
})

cv_nb
```

```
## $Fold01
##    accuracy sensitivity specificity
## 1 0.8804878   0.4931507     0.91834
##
## $Fold02
##    accuracy sensitivity specificity
## 1 0.8510131   0.4897959   0.8987854
##
## $Fold03
##    accuracy sensitivity specificity
## 1 0.8753117   0.4494382   0.9284712
##
## $Fold04
##    accuracy sensitivity specificity
## 1 0.8574794   0.4705882   0.9103079
##
## $Fold05
##   accuracy sensitivity specificity
## 1 0.867052   0.4210526   0.9220779
##
## $Fold06
##    accuracy sensitivity specificity
## 1 0.8445274   0.4166667   0.8944444
##
## $Fold07
##    accuracy sensitivity specificity
## 1 0.8639201   0.5048544   0.9169054
##
## $Fold08
##    accuracy sensitivity specificity
## 1 0.8573201   0.4835165   0.9048951
##
## $Fold09
##    accuracy sensitivity specificity
```

```
## 1 0.8587224   0.4333333   0.9116022
##
## $Fold10
##    accuracy sensitivity specificity
## 1 0.8607443         0.5   0.9110807
```

COMPARING ROC CURVE/AUC FOR ALL MODELS

```
plot(b_eval_lr, lty= 1, lwd=1, col = "red", colorize=F)
plot(b_eval_rf, lty=2, lwd= 2, col="green", add=TRUE)
plot(b_eval_nb, lty=3, lwd= 3, col="blue", add=TRUE)
abline(a=0, b=1)
legend(.6, .2, legend=c("LOGISTIC REGRESSION", "RANDOM FOREST","NAIVE BAYES"),
       col=c("red","green" ,"blue"), lty=1:3, cex=0.8)
```