CAC2\_41

Madhura KR

April 22, 2021

Topic: Employee Attrition

To predict if an employee is going to resign or not, using :**Classifcation algorithms- Random Forest and Logisitic Regression**

**Project description:**

Attrition is basically the turnover rate of employees inside an organization.It means loss of employee as well as talent. As employees leave an organization, they take with them much-need skills and qualifications that they develop during his tenure. It is costly from the company point of view to train the new employee.

The objective is to understand the influence of various variables on the Attrition rate, the degree of the influence and relation between the attributes. The main goal of this study is to implement a classification model for the dataset with good prediction results.

**Summary of the Dataset:**

For this study, IBM HR Data Analysis- Employee Attrition and Performance Attrition dataset has been selected, which has been obtained from Kaggle. It is a fictional dataset created by IBM data scientists, which contains data related to employee performance measures and attrition. The dataset contains several predictor variables , having varying influence on the response variable Attrition , which signifies whether an employee left the company or not.

The breakdown of the variable is as follows:

Response variable: Attrition (Yes or NO)

Predictor variables: 26 numerical, 6 categorical and 3 binary variables

Total instances: 1470

**MODEL 1 : RANDOM FOREST**

The Random Forest model has been built four times, in order to increase the model performances like Accuracy and Area Under the Curve. Way of increasing the model performances:

1. A random forest with just the pre procceesed data.

2. A random forest with tuning applied.

3. A random forest with feature engineered data

4. A random forest with feature engineered data and optimal number of trees

**MODEL 2- LOGISTIC REGRESSION**

As a way of increasing the model performance, three logistic models were built - 1. A Logistic Regression model with pre-processed data, removing the features causing multicollinearity.

1. A Logistic Regression model with just the significant features obtained using the anova() method and ‘chisq’ test as parameter.
2. A Logistic Regression model with 15 important features obtained using the Variable Importance plot.

The required packages:

suppressMessages(library(ggplot2))  
suppressMessages(library(grid))  
suppressMessages(library(gridExtra))  
#install.packages("mosaic")  
suppressMessages(library(mosaic))

## Warning: package 'mosaic' was built under R version 4.0.5

#install.packages("scales")  
suppressMessages(library(scales))

## Warning: package 'scales' was built under R version 4.0.5

suppressMessages(library(arules))

## Warning: package 'arules' was built under R version 4.0.5

suppressMessages(library(klaR))

## Warning: package 'klaR' was built under R version 4.0.5

suppressMessages(library(tictoc))  
suppressMessages(library(mlbench))

## Warning: package 'mlbench' was built under R version 4.0.5

suppressMessages(library(pROC))  
suppressMessages(library(plyr))  
suppressMessages(library(rpart))  
suppressMessages(library(rpart.plot))  
suppressMessages(library(randomForest))  
suppressMessages(library(caret))  
suppressMessages(library(ggplot2))  
suppressMessages(library(grid))  
suppressMessages(library(gridExtra))  
suppressMessages(library(dplyr))  
suppressMessages(library(rpart))  
suppressMessages(library(rpart.plot))  
suppressMessages(library(randomForest))  
suppressMessages(library(caret))  
#suppressMessages(library(gbm))  
suppressMessages(library(survival))  
suppressMessages(library(pROC))  
#suppressMessages(library(DMwR))  
suppressMessages(library(scales))  
suppressMessages(library(ROSE))  
suppressMessages(library(car))  
suppressMessages(library(tictoc))

library(readr)

##   
## Attaching package: 'readr'

## The following object is masked from 'package:scales':  
##   
## col\_factor

data<- read\_csv("C:/Users/madhu/Downloads/WA\_Fn-UseC\_-HR-Employee-Attrition.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## .default = col\_double(),  
## Attrition = col\_character(),  
## BusinessTravel = col\_character(),  
## Department = col\_character(),  
## EducationField = col\_character(),  
## Gender = col\_character(),  
## JobRole = col\_character(),  
## MaritalStatus = col\_character(),  
## Over18 = col\_character(),  
## OverTime = col\_character()  
## )  
## i Use `spec()` for the full column specifications.

head(data)

## # A tibble: 6 x 35  
## Age Attrition BusinessTravel DailyRate Department DistanceFromHome Education  
## <dbl> <chr> <chr> <dbl> <chr> <dbl> <dbl>  
## 1 41 Yes Travel\_Rarely 1102 Sales 1 2  
## 2 49 No Travel\_Freque~ 279 Research ~ 8 1  
## 3 37 Yes Travel\_Rarely 1373 Research ~ 2 2  
## 4 33 No Travel\_Freque~ 1392 Research ~ 3 4  
## 5 27 No Travel\_Rarely 591 Research ~ 2 1  
## 6 32 No Travel\_Freque~ 1005 Research ~ 2 2  
## # ... with 28 more variables: EducationField <chr>, EmployeeCount <dbl>,  
## # EmployeeNumber <dbl>, EnvironmentSatisfaction <dbl>, Gender <chr>,  
## # HourlyRate <dbl>, JobInvolvement <dbl>, JobLevel <dbl>, JobRole <chr>,  
## # JobSatisfaction <dbl>, MaritalStatus <chr>, MonthlyIncome <dbl>,  
## # MonthlyRate <dbl>, NumCompaniesWorked <dbl>, Over18 <chr>, OverTime <chr>,  
## # PercentSalaryHike <dbl>, PerformanceRating <dbl>,  
## # RelationshipSatisfaction <dbl>, StandardHours <dbl>,  
## # StockOptionLevel <dbl>, TotalWorkingYears <dbl>,  
## # TrainingTimesLastYear <dbl>, WorkLifeBalance <dbl>, YearsAtCompany <dbl>,  
## # YearsInCurrentRole <dbl>, YearsSinceLastPromotion <dbl>,  
## # YearsWithCurrManager <dbl>

dim(data)

## [1] 1470 35

names(data)

## [1] "Age" "Attrition"   
## [3] "BusinessTravel" "DailyRate"   
## [5] "Department" "DistanceFromHome"   
## [7] "Education" "EducationField"   
## [9] "EmployeeCount" "EmployeeNumber"   
## [11] "EnvironmentSatisfaction" "Gender"   
## [13] "HourlyRate" "JobInvolvement"   
## [15] "JobLevel" "JobRole"   
## [17] "JobSatisfaction" "MaritalStatus"   
## [19] "MonthlyIncome" "MonthlyRate"   
## [21] "NumCompaniesWorked" "Over18"   
## [23] "OverTime" "PercentSalaryHike"   
## [25] "PerformanceRating" "RelationshipSatisfaction"  
## [27] "StandardHours" "StockOptionLevel"   
## [29] "TotalWorkingYears" "TrainingTimesLastYear"   
## [31] "WorkLifeBalance" "YearsAtCompany"   
## [33] "YearsInCurrentRole" "YearsSinceLastPromotion"   
## [35] "YearsWithCurrManager"

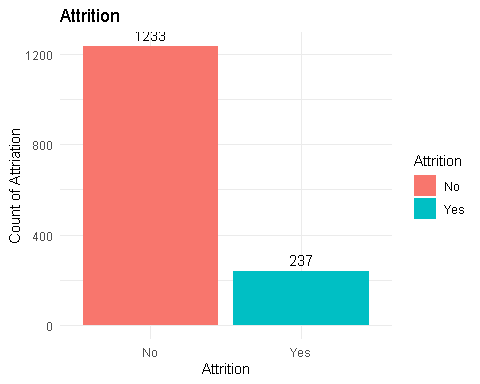
# checking missing values  
colSums(is.na(data))

## Age Attrition BusinessTravel   
## 0 0 0   
## DailyRate Department DistanceFromHome   
## 0 0 0   
## Education EducationField EmployeeCount   
## 0 0 0   
## EmployeeNumber EnvironmentSatisfaction Gender   
## 0 0 0   
## HourlyRate JobInvolvement JobLevel   
## 0 0 0   
## JobRole JobSatisfaction MaritalStatus   
## 0 0 0   
## MonthlyIncome MonthlyRate NumCompaniesWorked   
## 0 0 0   
## Over18 OverTime PercentSalaryHike   
## 0 0 0   
## PerformanceRating RelationshipSatisfaction StandardHours   
## 0 0 0   
## StockOptionLevel TotalWorkingYears TrainingTimesLastYear   
## 0 0 0   
## WorkLifeBalance YearsAtCompany YearsInCurrentRole   
## 0 0 0   
## YearsSinceLastPromotion YearsWithCurrManager   
## 0 0

There are no missing data values in the dataset.

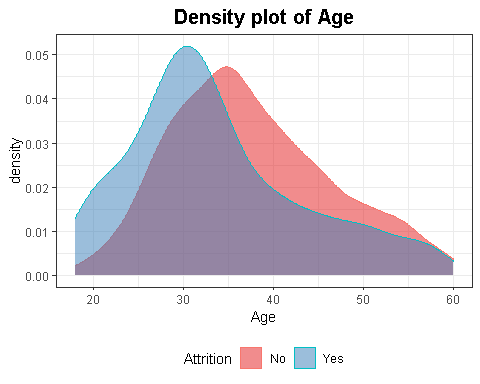
Visualizing the dataset- EDA

library(ggplot2)  
library(dplyr)  
data %>%  
 group\_by(Attrition) %>%  
 tally() %>%  
 ggplot(aes(x = Attrition, y = n,fill=Attrition)) +  
 geom\_bar(stat = "identity") +  
 theme\_minimal()+  
 labs(x="Attrition", y="Count of Attriation")+  
 ggtitle("Attrition")+  
 geom\_text(aes(label = n), vjust = -0.5, position = position\_dodge(0.9))

 In the plot above, class imbalance is clearly evident in the data.

Age Distribution by Attrition

ggplot(data, aes(x=Age, fill=Attrition, color=Attrition)) +  
 geom\_density(position="identity", alpha=0.5) +   
 theme\_bw() +   
 scale\_fill\_brewer(palette="Set1") +  
 ggtitle("Density plot of Age") +   
 theme(plot.title = element\_text(face = "bold", hjust = 0.5, size = 15), legend.position="bottom") +  
 labs(x = "Age")

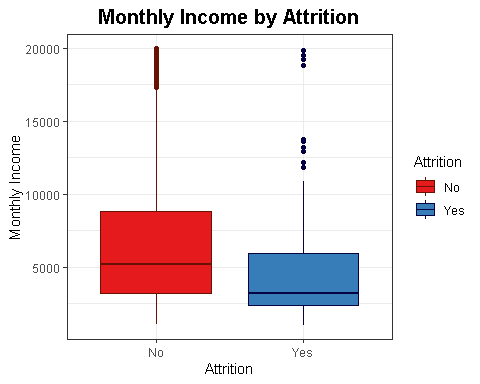
 The majority of employees are between 28-36 years. It seems to a large majority of those who left were relatively younger.

Monthly Income by Attrition

data %>% group\_by(Attrition) %>% summarize(Mean = mean(MonthlyIncome))

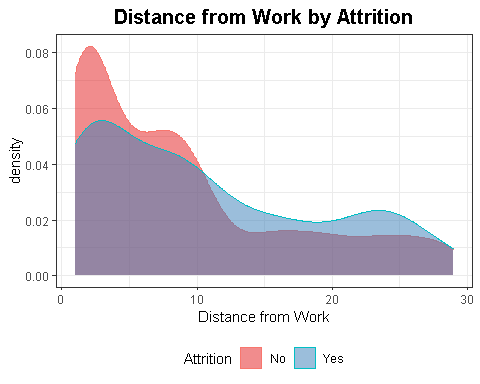
## Mean  
## 1 6502.931

ggplot(data, aes(x=Attrition, y=MonthlyIncome, color=Attrition, fill=Attrition)) +  
 geom\_boxplot() +   
 theme\_bw() +   
 scale\_fill\_brewer(palette="Set1") +  
 scale\_color\_manual(values=c("#661304", "#040242")) +  
 ggtitle("Monthly Income by Attrition") +   
 theme(plot.title = element\_text(face = "bold", hjust = 0.5, size = 15)) +  
 labs(x = "Attrition", y = "Monthly Income")

 It seems to a large majority of those who left had a relatively lower monthly income.

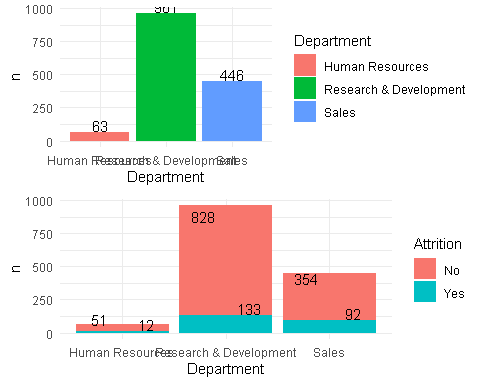
Distance from Work by Attrition

ggplot(data, aes(x=DistanceFromHome, fill=Attrition, color=Attrition)) +  
 geom\_density(position="identity", alpha=0.5) +   
 theme\_bw() +   
 scale\_fill\_brewer(palette="Set1") +  
 ggtitle("Distance from Work by Attrition") +   
 theme(plot.title = element\_text(face = "bold", hjust = 0.5, size = 15), legend.position="bottom") +  
 labs(x = "Distance from Work")

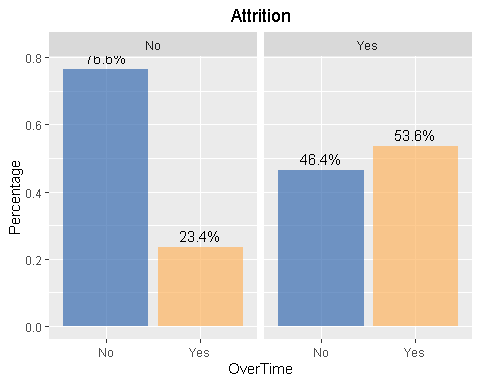


It seems to a large majority of those who left had a relatively lower distance from work.

g1 <- data %>%  
 group\_by(Department) %>%  
 tally() %>%  
 ggplot(aes(x = Department, y = n,fill=Department)) +  
 geom\_bar(stat = "identity") +  
 theme\_minimal()+  
 geom\_text(aes(label = n), vjust = -0.1, position = position\_dodge(0.9))  
  
g2 <- data %>%  
 group\_by(Department, Attrition) %>%  
 tally() %>%  
 ggplot(aes(x = Department, y = n,fill=Attrition)) +  
 geom\_bar(stat = "identity") +  
 theme\_minimal()+  
 geom\_text(aes(label = n), vjust = -0.1, position = position\_dodge(0.9))  
  
grid.arrange(g1,g2)

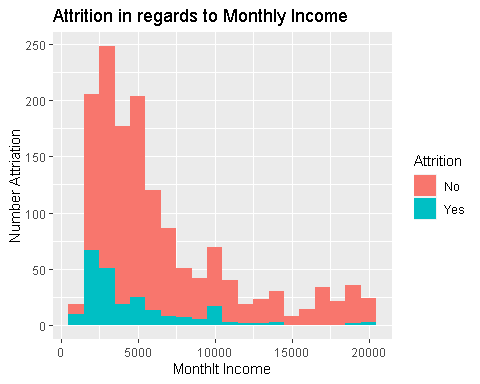
 Over Time by Attrition

ggplot(data,   
 aes(x = OverTime, group = Attrition)) +   
 geom\_bar(aes(y = ..prop.., fill = factor(..x..)),   
 stat="count",   
 alpha = 0.7) +  
 geom\_text(aes(label = scales::percent(..prop..), y = ..prop.. ),   
 stat= "count",   
 vjust = -.5) +  
 labs(y = "Percentage", fill= "OverTime") +  
 facet\_grid(~Attrition) +  
 scale\_fill\_manual(values = c("#386cb0","#fdb462")) +   
 theme(legend.position = "none", plot.title = element\_text(hjust = 0.5)) +   
 ggtitle("Attrition")

 There is a relatively higher percentage of people working overtime in the group of those who left. Also, while things seem to be going in the right direction for the group of people who still work for the IBM , the opposite is happening in the other group. It seems that there may be a pattern of people leaving because they are not promoted although they work hard. However,this is only an assumption at this point.

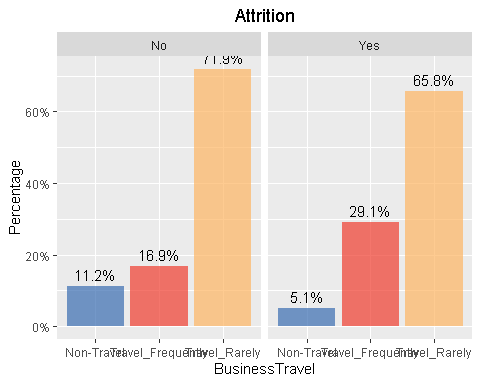
Monthly Income by Attrition

data %>%  
 ggplot(mapping = aes(x = MonthlyIncome)) +   
 geom\_histogram(aes(fill = Attrition), bins=20)+  
 labs(x="Monthlt Income", y="Number Attriation")+  
 ggtitle("Attrition in regards to Monthly Income")



Majority of the employees who left the company had income lesser than the average income of the employees of the same company.

ggplot(data,   
 aes(x= BusinessTravel, group=Attrition)) +   
 geom\_bar(aes(y = ..prop.., fill = factor(..x..)),   
 stat="count",   
 alpha = 0.7) +  
 geom\_text(aes(label = scales::percent(..prop..), y = ..prop.. ),   
 stat= "count",   
 vjust = -.5) +  
 labs(y = "Percentage", fill="Business Travel") +  
 facet\_grid(~Attrition) +  
 scale\_y\_continuous(labels=percent) +   
 scale\_fill\_manual(values = c("#386cb0","#ef3b2c", "#fdb462")) +   
 theme(legend.position = "none", plot.title = element\_text(hjust = 0.5)) +   
 ggtitle("Attrition")

 There seems to be a clear indication that those who left travelled more frequently compared to others. This might have also been an important reason behind their resignation.

Data Pre-processing: The dataset contains the variables EmployeeNumber, StandardHours, EmployeeCount, DailyRate, Over18 which do not add any new information to the dataset and hence was removed. Apart from this, There were no missing or nullvalues in the dataset. The outliers however were present, was not dealt with because of the reasons such as the data loss and less number of outliers.The string features were converted into categorical variables. Thus, the pre-proccessed data was obtained.

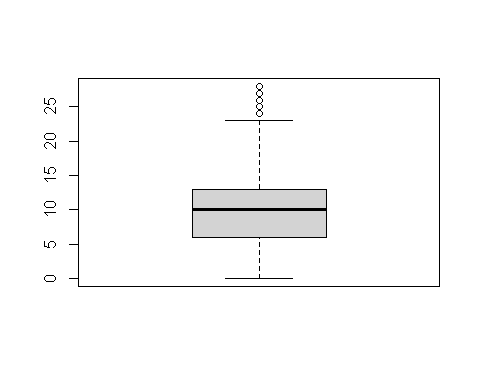
#Converting the character columns into categories  
character\_indices <- sapply(data, is.character)  
data[, character\_indices] <- lapply(data[,character\_indices], as.factor)  
data[, character\_indices]

## # A tibble: 1,470 x 9  
## Attrition BusinessTravel Department EducationField Gender JobRole   
## <fct> <fct> <fct> <fct> <fct> <fct>   
## 1 Yes Travel\_Rarely Sales Life Sciences Female Sales Execut~  
## 2 No Travel\_Frequent~ Research & De~ Life Sciences Male Research Sci~  
## 3 Yes Travel\_Rarely Research & De~ Other Male Laboratory T~  
## 4 No Travel\_Frequent~ Research & De~ Life Sciences Female Research Sci~  
## 5 No Travel\_Rarely Research & De~ Medical Male Laboratory T~  
## 6 No Travel\_Frequent~ Research & De~ Life Sciences Male Laboratory T~  
## 7 No Travel\_Rarely Research & De~ Medical Female Laboratory T~  
## 8 No Travel\_Rarely Research & De~ Life Sciences Male Laboratory T~  
## 9 No Travel\_Frequent~ Research & De~ Life Sciences Male Manufacturin~  
## 10 No Travel\_Rarely Research & De~ Medical Male Healthcare R~  
## # ... with 1,460 more rows, and 3 more variables: MaritalStatus <fct>,  
## # Over18 <fct>, OverTime <fct>

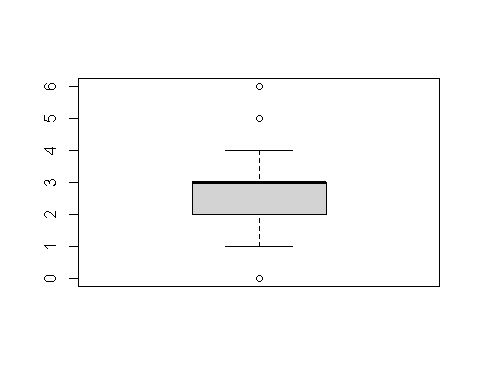
categorical\_indices <- sapply(data, is.factor)  
categorical\_indices

## Age Attrition BusinessTravel   
## FALSE TRUE TRUE   
## DailyRate Department DistanceFromHome   
## FALSE TRUE FALSE   
## Education EducationField EmployeeCount   
## FALSE TRUE FALSE   
## EmployeeNumber EnvironmentSatisfaction Gender   
## FALSE FALSE TRUE   
## HourlyRate JobInvolvement JobLevel   
## FALSE FALSE FALSE   
## JobRole JobSatisfaction MaritalStatus   
## TRUE FALSE TRUE   
## MonthlyIncome MonthlyRate NumCompaniesWorked   
## FALSE FALSE FALSE   
## Over18 OverTime PercentSalaryHike   
## TRUE TRUE FALSE   
## PerformanceRating RelationshipSatisfaction StandardHours   
## FALSE FALSE FALSE   
## StockOptionLevel TotalWorkingYears TrainingTimesLastYear   
## FALSE FALSE FALSE   
## WorkLifeBalance YearsAtCompany YearsInCurrentRole   
## FALSE FALSE FALSE   
## YearsSinceLastPromotion YearsWithCurrManager   
## FALSE FALSE

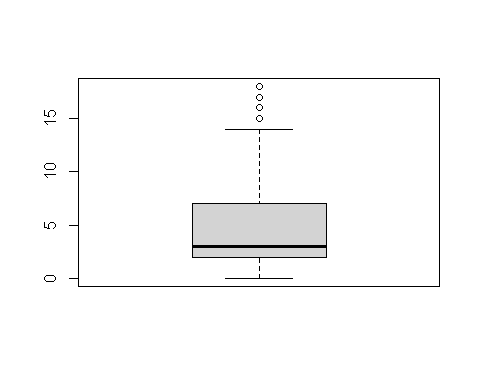
#To check the outliers using boxplot  
data$TotalWorkingYears[data$TotalWorkingYears %in% boxplot.stats(data$TotalWorkingYears)$out]<- median(data$TotalWorkingYears)  
boxplot(data$TotalWorkingYears)



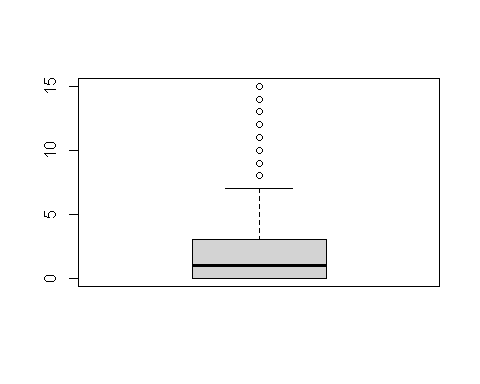
boxplot(data$TrainingTimesLastYear)



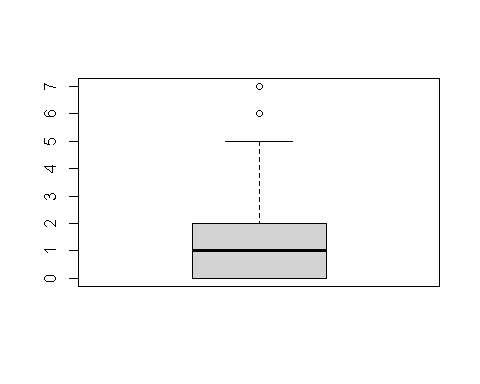
boxplot(data$YearsInCurrentRole)



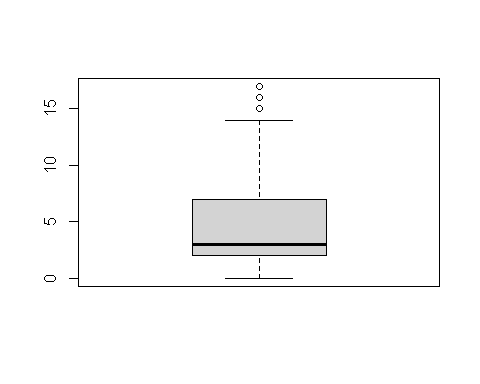
boxplot(data$YearsSinceLastPromotion)



data$YearsSinceLastPromotion[data$YearsSinceLastPromotion %in% boxplot.stats(data$YearsSinceLastPromotion)$out]<- median(data$YearsSinceLastPromotion)  
boxplot(data$YearsSinceLastPromotion)



boxplot(data$YearsWithCurrManager)



#Unnecessary columns through investigation  
#1. Employee Number - Considering it as a unique employee identification number. It would not add much value to analysis  
#2. StandardHours - Considering its value = 80 across all the rows in the dataset, it can be deleted.  
data$EmployeeNumber <- NULL  
data$StandardHours <- NULL  
data$EmployeeCount <- NULL  
data$DailyRate <- NULL  
data$Over18 <- NULL

dim(data)

## [1] 1470 30

#We have reduced the number of features from 35 to useful 30 features.

#Checking for NA/missing values  
sum(!complete.cases(data))

## [1] 0

#Shows the presence of 1470 complete data   
  
#Finding out the columns with NAs/missing values  
colSums(is.na(data))

## Age Attrition BusinessTravel   
## 0 0 0   
## Department DistanceFromHome Education   
## 0 0 0   
## EducationField EnvironmentSatisfaction Gender   
## 0 0 0   
## HourlyRate JobInvolvement JobLevel   
## 0 0 0   
## JobRole JobSatisfaction MaritalStatus   
## 0 0 0   
## MonthlyIncome MonthlyRate NumCompaniesWorked   
## 0 0 0   
## OverTime PercentSalaryHike PerformanceRating   
## 0 0 0   
## RelationshipSatisfaction StockOptionLevel TotalWorkingYears   
## 0 0 0   
## TrainingTimesLastYear WorkLifeBalance YearsAtCompany   
## 0 0 0   
## YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager   
## 0 0 0

#Shows that none of the columns have NA values  
  
  
#Checking for Duplicate entries  
nrow(data[!(duplicated(data)),])

## [1] 1470

nrow(data)

## [1] 1470

#Shows there are no duplicate entries and all the rows in the dataset are unique and individual data

num\_indices <- sapply(data, is.numeric)  
stats\_summary <- lapply(data[,num\_indices], summary)  
stats\_summary

## $Age  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 18.00 30.00 36.00 36.92 43.00 60.00   
##   
## $DistanceFromHome  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 7.000 9.193 14.000 29.000   
##   
## $Education  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.913 4.000 5.000   
##   
## $EnvironmentSatisfaction  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.722 4.000 4.000   
##   
## $HourlyRate  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 30.00 48.00 66.00 65.89 83.75 100.00   
##   
## $JobInvolvement  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 2.00 3.00 2.73 3.00 4.00   
##   
## $JobLevel  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 2.064 3.000 5.000   
##   
## $JobSatisfaction  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.729 4.000 4.000   
##   
## $MonthlyIncome  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1009 2911 4919 6503 8379 19999   
##   
## $MonthlyRate  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2094 8047 14236 14313 20462 26999   
##   
## $NumCompaniesWorked  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 1.000 2.000 2.693 4.000 9.000   
##   
## $PercentSalaryHike  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 11.00 12.00 14.00 15.21 18.00 25.00   
##   
## $PerformanceRating  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.000 3.000 3.000 3.154 3.000 4.000   
##   
## $RelationshipSatisfaction  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.712 4.000 4.000   
##   
## $StockOptionLevel  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 1.0000 0.7939 1.0000 3.0000   
##   
## $TotalWorkingYears  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 6.00 10.00 10.31 13.00 28.00   
##   
## $TrainingTimesLastYear  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.000 3.000 2.799 3.000 6.000   
##   
## $WorkLifeBalance  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.761 3.000 4.000   
##   
## $YearsAtCompany  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 3.000 5.000 7.008 9.000 40.000   
##   
## $YearsInCurrentRole  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.000 3.000 4.229 7.000 18.000   
##   
## $YearsSinceLastPromotion  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 1.00 1.45 2.00 7.00   
##   
## $YearsWithCurrManager  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.000 3.000 4.123 7.000 17.000

#Conversion of features into categorical type  
data$BusinessTravel<-as.factor(data$BusinessTravel)  
data$Department<-as.factor(data$Department)  
data$Education <- as.factor(data$Education)  
data$EducationField<-as.factor(data$EducationField)  
data$EnvironmentSatisfaction <- as.factor(data$EnvironmentSatisfaction)  
data$Gender<-as.factor(data$Gender)  
data$JobInvolvement <- as.factor(data$JobInvolvement)  
data$JobLevel <- as.factor(data$JobLevel)  
data$JobRole<-as.factor(data$JobRole)  
data$JobSatisfaction <- as.factor(data$JobSatisfaction)  
data$MaritalStatus<-as.factor(data$MaritalStatus)  
data$PerformanceRating <- as.factor(data$PerformanceRating)  
data$RelationshipSatisfaction <- as.factor(data$RelationshipSatisfaction)  
data$WorkLifeBalance <- as.factor(data$WorkLifeBalance)  
data$StockOptionLevel <- as.factor(data$StockOptionLevel)  
summary(data)

## Age Attrition BusinessTravel  
## Min. :18.00 No :1233 Non-Travel : 150   
## 1st Qu.:30.00 Yes: 237 Travel\_Frequently: 277   
## Median :36.00 Travel\_Rarely :1043   
## Mean :36.92   
## 3rd Qu.:43.00   
## Max. :60.00   
##   
## Department DistanceFromHome Education  
## Human Resources : 63 Min. : 1.000 1:170   
## Research & Development:961 1st Qu.: 2.000 2:282   
## Sales :446 Median : 7.000 3:572   
## Mean : 9.193 4:398   
## 3rd Qu.:14.000 5: 48   
## Max. :29.000   
##   
## EducationField EnvironmentSatisfaction Gender HourlyRate   
## Human Resources : 27 1:284 Female:588 Min. : 30.00   
## Life Sciences :606 2:287 Male :882 1st Qu.: 48.00   
## Marketing :159 3:453 Median : 66.00   
## Medical :464 4:446 Mean : 65.89   
## Other : 82 3rd Qu.: 83.75   
## Technical Degree:132 Max. :100.00   
##   
## JobInvolvement JobLevel JobRole JobSatisfaction  
## 1: 83 1:543 Sales Executive :326 1:289   
## 2:375 2:534 Research Scientist :292 2:280   
## 3:868 3:218 Laboratory Technician :259 3:442   
## 4:144 4:106 Manufacturing Director :145 4:459   
## 5: 69 Healthcare Representative:131   
## Manager :102   
## (Other) :215   
## MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked OverTime   
## Divorced:327 Min. : 1009 Min. : 2094 Min. :0.000 No :1054   
## Married :673 1st Qu.: 2911 1st Qu.: 8047 1st Qu.:1.000 Yes: 416   
## Single :470 Median : 4919 Median :14236 Median :2.000   
## Mean : 6503 Mean :14313 Mean :2.693   
## 3rd Qu.: 8379 3rd Qu.:20462 3rd Qu.:4.000   
## Max. :19999 Max. :26999 Max. :9.000   
##   
## PercentSalaryHike PerformanceRating RelationshipSatisfaction StockOptionLevel  
## Min. :11.00 3:1244 1:276 0:631   
## 1st Qu.:12.00 4: 226 2:303 1:596   
## Median :14.00 3:459 2:158   
## Mean :15.21 4:432 3: 85   
## 3rd Qu.:18.00   
## Max. :25.00   
##   
## TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany   
## Min. : 0.00 Min. :0.000 1: 80 Min. : 0.000   
## 1st Qu.: 6.00 1st Qu.:2.000 2:344 1st Qu.: 3.000   
## Median :10.00 Median :3.000 3:893 Median : 5.000   
## Mean :10.31 Mean :2.799 4:153 Mean : 7.008   
## 3rd Qu.:13.00 3rd Qu.:3.000 3rd Qu.: 9.000   
## Max. :28.00 Max. :6.000 Max. :40.000   
##   
## YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager  
## Min. : 0.000 Min. :0.00 Min. : 0.000   
## 1st Qu.: 2.000 1st Qu.:0.00 1st Qu.: 2.000   
## Median : 3.000 Median :1.00 Median : 3.000   
## Mean : 4.229 Mean :1.45 Mean : 4.123   
## 3rd Qu.: 7.000 3rd Qu.:2.00 3rd Qu.: 7.000   
## Max. :18.000 Max. :7.00 Max. :17.000   
##

**MODEL 1 : RANDOM FOREST**

**The Random Forest model has been built four times, in order to increase the model performances like Accuracyand Area Under the Curve. Way of increasing the model performances:**

**1. A random forest with just the pre proceesed data.**

**2. A random forest with tuning applied.**

**3. A random forest with feature engineered data**

**4. A random forest with feature engineered data and optimal number of trees**

**Random Forest (preproccessed data)**

#Train-test splitting  
rfData1 <- data#to keep a copy of the raw data  
set.seed(123)  
indexes = sample(1:nrow(rfData1), size=0.7\*nrow(rfData1))  
RF\_train <- rfData1[indexes,]  
RF\_test <- rfData1[-indexes,]

Dealing with class imbalance

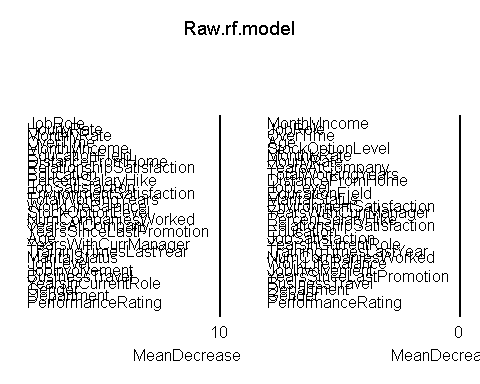
prop.table(table(RF\_train$Attrition))

##   
## No Yes   
## 0.8425656 0.1574344

train\_balanced1<- ovun.sample(Attrition~ ., data = RF\_train, method = "both", p=0.5, seed = 1)$data  
prop.table(table(train\_balanced1$Attrition))

##   
## No Yes   
## 0.5179786 0.4820214

#Model fitting and Variable importance  
tic('Model1 (Just preprocessed data)')  
Raw.rf.model <- randomForest(Attrition~.,data=train\_balanced1, importance=TRUE)  
varImpPlot(Raw.rf.model)



toc()

## Model1 (Just preprocessed data): 5.17 sec elapsed

Top five important features in the raw data : J**obRole, MonthlyRate, MonthlyIncome, HourlyRate and EducationField.**

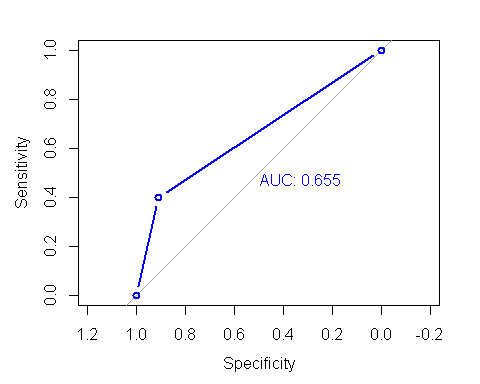
Raw.rf.prd <- predict(Raw.rf.model, newdata = RF\_test)  
confusionMatrix(RF\_test$Attrition, Raw.rf.prd)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 333 33  
## Yes 45 30  
##   
## Accuracy : 0.8231   
## 95% CI : (0.7843, 0.8576)  
## No Information Rate : 0.8571   
## P-Value [Acc > NIR] : 0.9803   
##   
## Kappa : 0.3309   
##   
## Mcnemar's Test P-Value : 0.2129   
##   
## Sensitivity : 0.8810   
## Specificity : 0.4762   
## Pos Pred Value : 0.9098   
## Neg Pred Value : 0.4000   
## Prevalence : 0.8571   
## Detection Rate : 0.7551   
## Detection Prevalence : 0.8299   
## Balanced Accuracy : 0.6786   
##   
## 'Positive' Class : No   
##

Raw.rf.plot<- plot.roc(as.numeric(RF\_test$Attrition), as.numeric(Raw.rf.prd),lwd=2, type="b",print.auc=TRUE,col ="blue")

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

 Random Forest (Pre-proccessed data): Accuracy= 0.8277 AUC= 0.663 Time = 2.65 sec elapsed

#Model 2 - Random Forest Model (with Tuning)  
tic('RF2-Tuning')  
#Define the cross validation  
tr\_control <- trainControl(method = 'cv', number = 5, classProbs = T)  
tunegrid <- expand.grid(.mtry=c(1:15))  
RF\_model\_2 = suppressWarnings(train(Attrition~., data=train\_balanced1  
 , method="rf"  
 , metric="ROC"  
 , tuneLength = 10  
 , tuneGrid = tunegrid  
 , trControl=tr\_control))  
  
toc()

## RF2-Tuning: 174.54 sec elapsed

#Predict using the Random Forest Model 2  
predict\_RF2 <- suppressWarnings( predict(RF\_model\_2,newdata = RF\_test,type = 'raw'))  
#Evaluating the RF model 2  
  
confusionMatrix(as.factor(predict\_RF2), as.factor(RF\_test$Attrition))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 335 46  
## Yes 31 29  
##   
## Accuracy : 0.8254   
## 95% CI : (0.7867, 0.8597)  
## No Information Rate : 0.8299   
## P-Value [Acc > NIR] : 0.6291   
##   
## Kappa : 0.328   
##   
## Mcnemar's Test P-Value : 0.1106   
##   
## Sensitivity : 0.9153   
## Specificity : 0.3867   
## Pos Pred Value : 0.8793   
## Neg Pred Value : 0.4833   
## Prevalence : 0.8299   
## Detection Rate : 0.7596   
## Detection Prevalence : 0.8639   
## Balanced Accuracy : 0.6510   
##   
## 'Positive' Class : No   
##

auc(as.numeric(RF\_test$Attrition), as.numeric(predict\_RF2))

## Setting levels: control = 1, case = 2

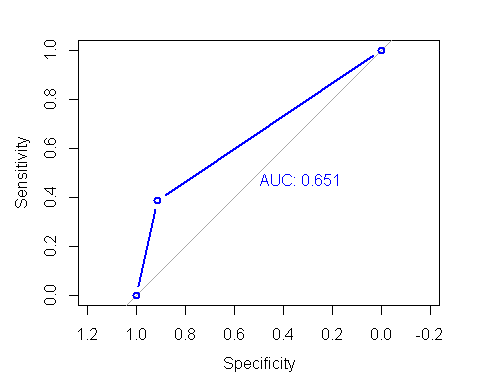
## Setting direction: controls < cases

## Area under the curve: 0.651

rf.Plot2<- plot.roc (as.numeric(RF\_test$Attrition), as.numeric(predict\_RF2),lwd=2, type="b", print.auc=TRUE,col ="blue")

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

 Random forest(Tuning): Accuracy = 0.8254 AUC = 0.667 Time = 147.42 sec elapsed

**FEATURE ENGINEERING**

**Now we want to use some data wrapping to make the results better:**

Making Age Group 18-24 = Young , 25-54=Middle-Age , 54-120= Adult

pre\_data<-data  
pre\_data$AgeGroup <- as.factor(ifelse(pre\_data$Age<=24,"Young", ifelse(pre\_data$Age<=54,"Middle-Age","Adult")))  
   
table(pre\_data$AgeGroup)

##   
## Adult Middle-Age Young   
## 69 1304 97

As it can be seen, the majority of employees are middle aged.

Less or more than average Monthly Income We calculate the average income and generate the level of incom(High or Low)

pre\_data$IncomeLevel <- as.factor(  
 ifelse(pre\_data$MonthlyIncome<ave(pre\_data$MonthlyIncome),"Low","High")  
)  
table(pre\_data$IncomeLevel)

##   
## High Low   
## 493 977

#Investigate the numerical columns  
all\_indices <- sapply(pre\_data, is.numeric)  
str(pre\_data[, all\_indices])

## tibble [1,470 x 13] (S3: tbl\_df/tbl/data.frame)  
## $ Age : num [1:1470] 41 49 37 33 27 32 59 30 38 36 ...  
## $ DistanceFromHome : num [1:1470] 1 8 2 3 2 2 3 24 23 27 ...  
## $ HourlyRate : num [1:1470] 94 61 92 56 40 79 81 67 44 94 ...  
## $ MonthlyIncome : num [1:1470] 5993 5130 2090 2909 3468 ...  
## $ MonthlyRate : num [1:1470] 19479 24907 2396 23159 16632 ...  
## $ NumCompaniesWorked : num [1:1470] 8 1 6 1 9 0 4 1 0 6 ...  
## $ PercentSalaryHike : num [1:1470] 11 23 15 11 12 13 20 22 21 13 ...  
## $ TotalWorkingYears : num [1:1470] 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : num [1:1470] 0 3 3 3 3 2 3 2 2 3 ...  
## $ YearsAtCompany : num [1:1470] 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : num [1:1470] 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion: num [1:1470] 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : num [1:1470] 5 7 0 0 2 6 0 0 8 7 ...

It can be observed that the columns - MonthlyIncome, MOnthlyRate contain significant and distinct numerical values. These cannot be be directly converted into factors because it would result in unnecessarily large number of levels. Hence, we discretize them so that manageable number of columns are obtained.

pre\_data$MonthlyIncome <- discretize(pre\_data$MonthlyIncome, method = 'interval')  
pre\_data$MonthlyRate <- discretize(pre\_data$MonthlyRate, method = 'interval', categories = 4)

## Warning in discretize(pre\_data$MonthlyRate, method = "interval", categories =  
## 4): Parameter categories is deprecated. Use breaks instead! Also, the default  
## method is now frequency!

str(pre\_data[, all\_indices])

## tibble [1,470 x 13] (S3: tbl\_df/tbl/data.frame)  
## $ Age : num [1:1470] 41 49 37 33 27 32 59 30 38 36 ...  
## $ DistanceFromHome : num [1:1470] 1 8 2 3 2 2 3 24 23 27 ...  
## $ HourlyRate : num [1:1470] 94 61 92 56 40 79 81 67 44 94 ...  
## $ MonthlyIncome : Factor w/ 3 levels "[1.01e+03,7.34e+03)",..: 1 1 1 1 1 1 1 1 2 1 ...  
## ..- attr(\*, "discretized:breaks")= num [1:4] 1009 7339 13669 19999  
## ..- attr(\*, "discretized:method")= chr "interval"  
## $ MonthlyRate : Factor w/ 4 levels "[2.09e+03,8.32e+03)",..: 3 4 1 4 3 2 2 2 2 3 ...  
## ..- attr(\*, "discretized:breaks")= num [1:5] 2094 8320 14546 20773 26999  
## ..- attr(\*, "discretized:method")= chr "interval"  
## $ NumCompaniesWorked : num [1:1470] 8 1 6 1 9 0 4 1 0 6 ...  
## $ PercentSalaryHike : num [1:1470] 11 23 15 11 12 13 20 22 21 13 ...  
## $ TotalWorkingYears : num [1:1470] 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : num [1:1470] 0 3 3 3 3 2 3 2 2 3 ...  
## $ YearsAtCompany : num [1:1470] 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : num [1:1470] 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion: num [1:1470] 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : num [1:1470] 5 7 0 0 2 6 0 0 8 7 ...

#Retaining the columns that were discretized and removing the corresponding base columns  
pre\_data$Age<-NULL  
pre\_data$MonthlyIncome<-NULL

pre\_data$HourlyRate<-NULL

dim(pre\_data)

## [1] 1470 29

The dimensionality has been reduced to 29 columns.

NEW RANDOM FOREST

rfData <- pre\_data  
set.seed(123)  
indexes = sample(1:nrow(rfData), size=0.7\*nrow(rfData))  
RFtrain.Data <- rfData[indexes,]  
RFtest.Data <- rfData[-indexes,]

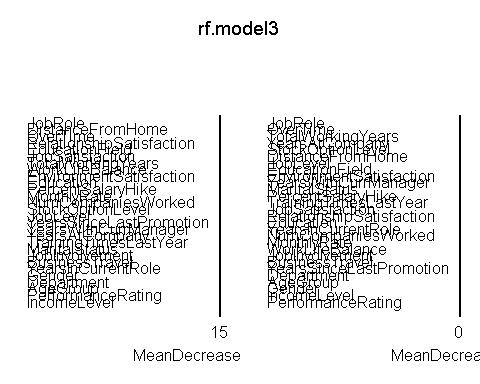
prop.table(table(RFtrain.Data$Attrition))

##   
## No Yes   
## 0.8425656 0.1574344

train\_balanced<- ovun.sample(Attrition~ ., data = RFtrain.Data, method = "both", p=0.5, seed = 1)$data  
#data\_balanced<-data.frame(data\_balanced)  
prop.table(table(train\_balanced$Attrition))

##   
## No Yes   
## 0.5179786 0.4820214

tic('RF3\_featureEng')  
rf.model3 <- randomForest(Attrition~.,train\_balanced, importance=TRUE)  
varImpPlot(rf.model3)



toc()

## RF3\_featureEng: 4.14 sec elapsed

From, the Mean Decrease Accuracy plot, the top 5 variables are JobRole,DistanceFromHome, OverTime, Education and TotalWorkingYears

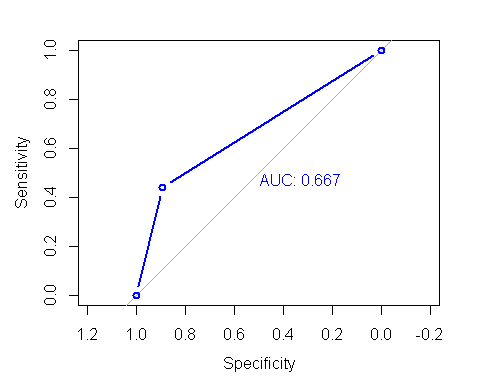
rf.prd <- predict(rf.model3, newdata = RFtest.Data)  
confusionMatrix(RFtest.Data$Attrition, rf.prd)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 327 39  
## Yes 42 33  
##   
## Accuracy : 0.8163   
## 95% CI : (0.777, 0.8514)  
## No Information Rate : 0.8367   
## P-Value [Acc > NIR] : 0.8883   
##   
## Kappa : 0.3388   
##   
## Mcnemar's Test P-Value : 0.8241   
##   
## Sensitivity : 0.8862   
## Specificity : 0.4583   
## Pos Pred Value : 0.8934   
## Neg Pred Value : 0.4400   
## Prevalence : 0.8367   
## Detection Rate : 0.7415   
## Detection Prevalence : 0.8299   
## Balanced Accuracy : 0.6723   
##   
## 'Positive' Class : No   
##

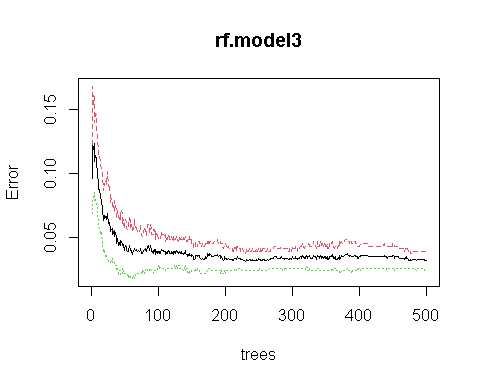
rf.Plot3<- plot.roc (as.numeric(RFtest.Data$Attrition), as.numeric(rf.prd),lwd=2, type="b", print.auc=TRUE,col ="blue")

## Setting levels: control = 1, case = 2

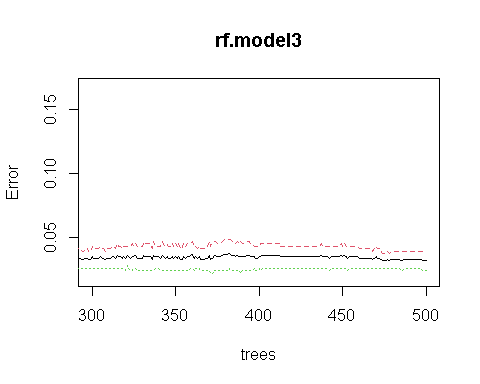
## Setting direction: controls < cases

 Random Forest(After Feature Engineering): Accuracy = 0.8299 AUC = 0.686 Time = 2.26 sec elapsed

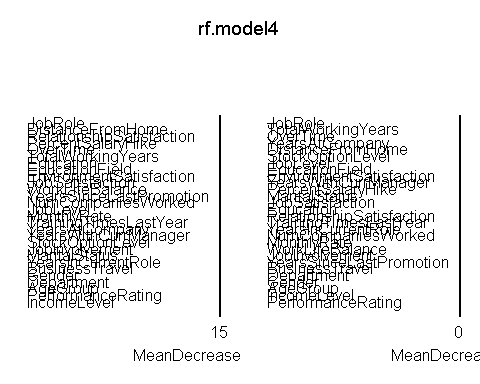
options(repr.plot.width=6, repr.plot.height=4)  
plot(rf.model3)



#Zooming in  
options(repr.plot.width=7, repr.plot.height=4)  
plot(rf.model3, xlim=c(300,500))

 500 or 550 is the optimal number of trees in this case.

tic('RF3\_featureEng+Optimal-ntree')  
rf.model4 <- randomForest(Attrition~.,train\_balanced, importance=TRUE,ntree=550)  
varImpPlot(rf.model4)



toc()

## RF3\_featureEng+Optimal-ntree: 4.61 sec elapsed

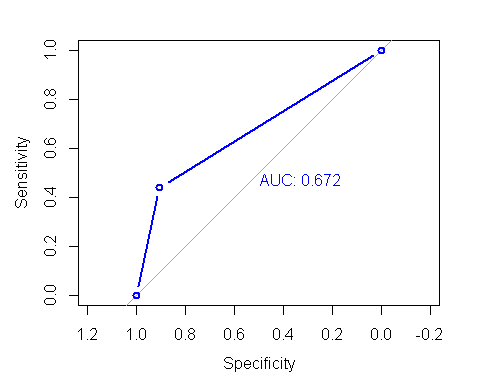
rf.prd1 <- predict(rf.model4, newdata = RFtest.Data)  
confusionMatrix(RFtest.Data$Attrition, rf.prd1)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 331 35  
## Yes 42 33  
##   
## Accuracy : 0.8254   
## 95% CI : (0.7867, 0.8597)  
## No Information Rate : 0.8458   
## P-Value [Acc > NIR] : 0.8934   
##   
## Kappa : 0.3576   
##   
## Mcnemar's Test P-Value : 0.4941   
##   
## Sensitivity : 0.8874   
## Specificity : 0.4853   
## Pos Pred Value : 0.9044   
## Neg Pred Value : 0.4400   
## Prevalence : 0.8458   
## Detection Rate : 0.7506   
## Detection Prevalence : 0.8299   
## Balanced Accuracy : 0.6863   
##   
## 'Positive' Class : No   
##

rf.Plot4<- plot.roc (as.numeric(RFtest.Data$Attrition), as.numeric(rf.prd1),lwd=2, type="b", print.auc=TRUE,col ="blue")

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases



Random Forest (with feature engineering and optimal number of trees) : Accuracy =0.8322 AUC = 0.676 Time = 2.67 sec elapsed

par(mfrow=c(2,2))  
rf.Plot1<- plot.roc(as.numeric(RF\_test$Attrition), as.numeric(Raw.rf.prd),lwd=2,print.auc=TRUE,col ="blue",main="Random Forest(Just Pre-processes data)", type="b")

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

rf.Plot2<- plot.roc (as.numeric(RF\_test$Attrition), as.numeric(predict\_RF2), print.auc=TRUE,col ="blue",main="Random Forest(Tuning)", type="b")

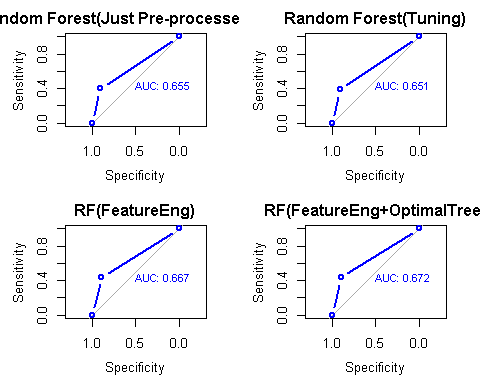
## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases

rf.Plot3<- plot.roc (as.numeric(RFtest.Data$Attrition), as.numeric(rf.prd), print.auc=TRUE,col ="blue",main="RF(FeatureEng)",type="b")

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases

rf.Plot4<- plot.roc (as.numeric(RFtest.Data$Attrition), as.numeric(rf.prd1), print.auc=TRUE,col ="blue",main="RF(FeatureEng+OptimalTree)",type="b")

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases



Method<-c('Random Forest(Just Pre-proccessed data)','RF(Tuning)','RF(FeatureEngineering)','RF(FeatureEng+Optimal\_ntree)')  
acc<-c(0.8277, 0.8254,0.8299,0.8322)  
compare<-data.frame(Method,acc)  
compare

## Method acc  
## 1 Random Forest(Just Pre-proccessed data) 0.8277  
## 2 RF(Tuning) 0.8254  
## 3 RF(FeatureEngineering) 0.8299  
## 4 RF(FeatureEng+Optimal\_ntree) 0.8322

**Inferences:**

The Random forest model with just the pre proccessed data, with the class imbalance issue solved, gave an accuracy of 82.77 %.

The model when tuned with tunegrid and traincontrol(), resulted in a poor model performance with a decreased accuracy of 82.54%.

So, some data wrapping was implemented to make the results better. The discretized variables were retained and the original features were removed. The model implementation with this data, improved the model performance giving an accuracy of 82.99%. The same model was checked with the parameter ntree.

With the additon of the optimal number of trees (550) in the model, the accuracy further increased to 83.22%. Thus, the best random forest model has been obtained with an accuracy of 83.22% and AUC score of 0.676 (which is the highest among the AUCs). The Time Complexity of the different model building were same, except that the tuning of the model took more time.(noted in the results provided).

**LOGISTIC REGRESSION**

**As a way of incresing the model performance, three logistic models were built - 1. A Logistic Regression model with pre-processed data, removing the features causing multicollinearity.**

1. **A Logistic Regression model with just the significant features obtained using the anova() method and ‘chisq’ test as parameter.**
2. **A Logistic Regression model with 15 important features obtained using the Variable Importance plot.**

set.seed(123)  
logdata <- pre\_data  
logdata$Attrition <- mapvalues(logdata$Attrition, from=c('Yes', 'No'), to = c(1,0))  
logdata$Attrition<-as.factor(logdata$Attrition)  
indexes <- sample(1:nrow(logdata), size=0.7\*nrow(logdata))  
logtrain.Data <- logdata[indexes,]  
logtest.Data <- logdata[-indexes,]

prop.table(table(logtrain.Data$Attrition))

##   
## 0 1   
## 0.8425656 0.1574344

library(ROSE)  
train\_balanced3<- ovun.sample(Attrition~ ., data = logtrain.Data, method = "both", p=0.5, seed = 1)$data  
#data\_balanced<-data.frame(data\_balanced)  
prop.table(table(train\_balanced3$Attrition))

##   
## 0 1   
## 0.5179786 0.4820214

numerics <- unlist(lapply(train\_balanced3, is.numeric))  
numerics <- train\_balanced3[,numerics]

dummy\_model <- glm(train\_balanced3$Attrition~., data = numerics, family = "binomial")  
print(vif(dummy\_model))

## DistanceFromHome NumCompaniesWorked PercentSalaryHike   
## 1.036784 1.265639 1.026251   
## TotalWorkingYears TrainingTimesLastYear YearsAtCompany   
## 1.779274 1.018546 3.521571   
## YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager   
## 3.180624 1.261518 2.941924

YearsAtCompany has its VIF value greater than 5. It will be removed from the data.

train\_balanced3 <- subset(train\_balanced3, select = - c(YearsAtCompany))  
logtest.Data<- subset(logtest.Data, select = - c(YearsAtCompany))

numerics <- unlist(lapply(train\_balanced3, is.numeric))  
numerics <- train\_balanced3[,numerics]  
dummy\_model <- glm(train\_balanced3$Attrition~., data = numerics, family = "binomial")  
print(vif(dummy\_model))

## DistanceFromHome NumCompaniesWorked PercentSalaryHike   
## 1.036100 1.246833 1.017694   
## TotalWorkingYears TrainingTimesLastYear YearsInCurrentRole   
## 1.712002 1.018059 2.457321   
## YearsSinceLastPromotion YearsWithCurrManager   
## 1.253779 2.426472

#Model fitting  
model1 <- glm(Attrition ~.,family=binomial(link="logit"),data=train\_balanced3)

#Predict using the XGBM Model   
predict\_log1 <- suppressWarnings(  
 predict(model1,newdata = logtest.Data))  
predict\_log1<-ifelse(predict\_log1>0.5,1,0)  
predict\_log1<-as.factor(predict\_log1)  
  
#Evaluating the model  
confusionMatrix(as.factor(predict\_log1), as.factor(logtest.Data$Attrition))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 300 23  
## 1 66 52  
##   
## Accuracy : 0.7982   
## 95% CI : (0.7576, 0.8347)  
## No Information Rate : 0.8299   
## P-Value [Acc > NIR] : 0.9647   
##   
## Kappa : 0.4178   
##   
## Mcnemar's Test P-Value : 8.508e-06   
##   
## Sensitivity : 0.8197   
## Specificity : 0.6933   
## Pos Pred Value : 0.9288   
## Neg Pred Value : 0.4407   
## Prevalence : 0.8299   
## Detection Rate : 0.6803   
## Detection Prevalence : 0.7324   
## Balanced Accuracy : 0.7565   
##   
## 'Positive' Class : 0   
##

log\_auc<-auc(as.numeric(logtest.Data$Attrition), as.numeric(predict\_log1))

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

log\_auc

## Area under the curve: 0.7565

Model 2- Selecting significant variables using anova() and ‘chisq’ test (10% LOS)

# Feature analysis   
# Based upon the p-value of anova  
anova(model1, test = "Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: Attrition  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 1028 1425.17   
## BusinessTravel 2 44.819 1026 1380.35 1.852e-10 \*\*\*  
## Department 2 10.067 1024 1370.28 0.0065167 \*\*   
## DistanceFromHome 1 6.803 1023 1363.48 0.0091007 \*\*   
## Education 4 3.340 1019 1360.14 0.5025673   
## EducationField 5 20.598 1014 1339.54 0.0009647 \*\*\*  
## EnvironmentSatisfaction 3 30.080 1011 1309.46 1.328e-06 \*\*\*  
## Gender 1 0.519 1010 1308.94 0.4714323   
## JobInvolvement 3 22.303 1007 1286.64 5.640e-05 \*\*\*  
## JobLevel 4 127.406 1003 1159.23 < 2.2e-16 \*\*\*  
## JobRole 8 30.806 995 1128.43 0.0001521 \*\*\*  
## JobSatisfaction 3 24.179 992 1104.25 2.292e-05 \*\*\*  
## MaritalStatus 2 64.647 990 1039.60 9.165e-15 \*\*\*  
## MonthlyRate 3 8.580 987 1031.02 0.0354315 \*   
## NumCompaniesWorked 1 8.211 986 1022.81 0.0041631 \*\*   
## OverTime 1 85.747 985 937.06 < 2.2e-16 \*\*\*  
## PercentSalaryHike 1 0.043 984 937.02 0.8361509   
## PerformanceRating 1 5.476 983 931.54 0.0192756 \*   
## RelationshipSatisfaction 3 33.384 980 898.16 2.673e-07 \*\*\*  
## StockOptionLevel 3 9.026 977 889.13 0.0289460 \*   
## TotalWorkingYears 1 18.703 976 870.43 1.527e-05 \*\*\*  
## TrainingTimesLastYear 1 5.570 975 864.86 0.0182711 \*   
## WorkLifeBalance 3 24.910 972 839.95 1.612e-05 \*\*\*  
## YearsInCurrentRole 1 3.555 971 836.39 0.0593646 .   
## YearsSinceLastPromotion 1 24.118 970 812.28 9.061e-07 \*\*\*  
## YearsWithCurrManager 1 7.136 969 805.14 0.0075553 \*\*   
## AgeGroup 2 18.331 967 786.81 0.0001046 \*\*\*  
## IncomeLevel 1 0.461 966 786.35 0.4971440   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

train\_balanced3 <- subset(train\_balanced3, select = - c(Education))  
logtest.Data<- subset(logtest.Data, select = - c(Education))  
  
train\_balanced3 <- subset(train\_balanced3, select = - c(Gender))  
logtest.Data<- subset(logtest.Data, select = - c(Gender))  
  
train\_balanced3 <- subset(train\_balanced3, select = - c(PercentSalaryHike))  
logtest.Data<- subset(logtest.Data, select = - c(PercentSalaryHike))

# Fitting the Logistic Regression Model  
logmodel <- glm(Attrition ~., family=binomial(link="logit"), data =train\_balanced3)

predict\_log2 <- suppressWarnings(  
 predict(logmodel,newdata = logtest.Data))  
predict\_log2<-ifelse(predict\_log2>0.5,1,0)  
predict\_log2<-as.factor(predict\_log2)  
  
#Evaluating the model  
confusionMatrix(as.factor(predict\_log2), as.factor(logtest.Data$Attrition))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 302 22  
## 1 64 53  
##   
## Accuracy : 0.805   
## 95% CI : (0.7649, 0.8409)  
## No Information Rate : 0.8299   
## P-Value [Acc > NIR] : 0.9255   
##   
## Kappa : 0.435   
##   
## Mcnemar's Test P-Value : 9.818e-06   
##   
## Sensitivity : 0.8251   
## Specificity : 0.7067   
## Pos Pred Value : 0.9321   
## Neg Pred Value : 0.4530   
## Prevalence : 0.8299   
## Detection Rate : 0.6848   
## Detection Prevalence : 0.7347   
## Balanced Accuracy : 0.7659   
##   
## 'Positive' Class : 0   
##

log\_auc2<-auc(as.numeric(logtest.Data$Attrition), as.numeric(predict\_log2))

## Setting levels: control = 1, case = 2

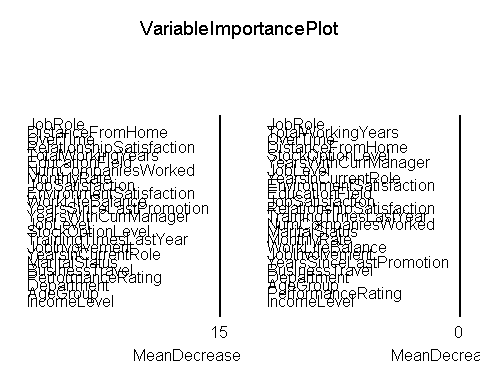
## Setting direction: controls < cases

log\_auc2

## Area under the curve: 0.7659

Model 3- Selecting important variables using VariableImportancePlot

set.seed(823)  
VariableImportancePlot <- randomForest(as.factor(Attrition) ~. , data = train\_balanced3, importance=TRUE)  
varImpPlot(VariableImportancePlot)



train3<-subset(train\_balanced3,select=-c(YearsWithCurrManager,TrainingTimesLastYear,MaritalStatus,YearsInCurrentRole,JobInvolvement,BusinessTravel,Department,AgeGroup,PerformanceRating))  
test3<-subset(logtest.Data,select=-c(YearsWithCurrManager,TrainingTimesLastYear,MaritalStatus,YearsInCurrentRole,JobInvolvement,BusinessTravel,Department,AgeGroup,PerformanceRating))

finalGlm <- glm(Attrition~., data = train3, family = "binomial")

predict\_log3<- suppressWarnings(  
 predict(finalGlm,newdata = test3))  
predict\_log3<-ifelse(predict\_log3>0.5,1,0)  
predict\_log3<-as.factor(predict\_log3)  
  
#Evaluating the model  
confusionMatrix(as.factor(predict\_log3), as.factor(test3$Attrition))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 306 20  
## 1 60 55  
##   
## Accuracy : 0.8186   
## 95% CI : (0.7794, 0.8535)  
## No Information Rate : 0.8299   
## P-Value [Acc > NIR] : 0.7594   
##   
## Kappa : 0.4698   
##   
## Mcnemar's Test P-Value : 1.299e-05   
##   
## Sensitivity : 0.8361   
## Specificity : 0.7333   
## Pos Pred Value : 0.9387   
## Neg Pred Value : 0.4783   
## Prevalence : 0.8299   
## Detection Rate : 0.6939   
## Detection Prevalence : 0.7392   
## Balanced Accuracy : 0.7847   
##   
## 'Positive' Class : 0   
##

log\_auc3<-auc(as.numeric(test3$Attrition), as.numeric(predict\_log3))

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

log\_auc3

## Area under the curve: 0.7847

Model<-c('Logistic(Just Pre-proccessed data)','Log(significant)','Log(important)')  
Accuracy<-c(0.7959,0.7868,0.8095)  
comparing<-data.frame(Model,Accuracy)  
comparing

## Model Accuracy  
## 1 Logistic(Just Pre-proccessed data) 0.7959  
## 2 Log(significant) 0.7868  
## 3 Log(important) 0.8095

par(mfrow=c(2,2))  
lg.Plot1<- plot.roc(as.numeric(logtest.Data$Attrition), as.numeric(predict\_log1),lwd=2,print.auc=TRUE,col ="blue",main="Logistic(Just Pre-processes data)", type="b")

## Setting levels: control = 1, case = 2

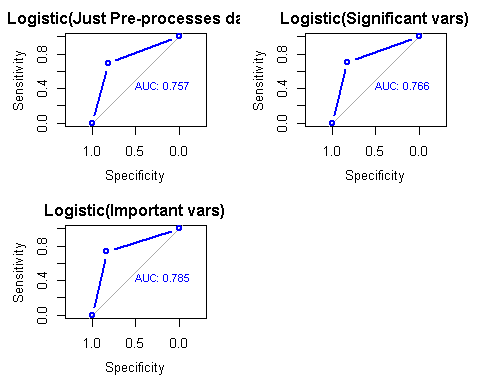
## Setting direction: controls < cases

lg.Plot2<- plot.roc (as.numeric(logtest.Data$Attrition), as.numeric(predict\_log2), print.auc=TRUE,col ="blue",main="Logistic(Significant vars)", type="b")

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases

lg.Plot3<- plot.roc (as.numeric(test3$Attrition), as.numeric(predict\_log3), print.auc=TRUE,col ="blue",main="Logistic(Important vars)",type="b")

## Setting levels: control = 1, case = 2  
## Setting direction: controls < cases



The logistic regression model with the pre proccessed was checked for multicolinearity.The variable ‘ÝearsAtCompany’, which had VIF of above 5 was removed. The model was further fitted which gave an accuracy of 79.59%.

In attempt of increasing the accuracy and AUC, only sigificant variables (10 % LOS) were considered as predictors. But, the result proved otherwise. The accuracy was decreased to 78.68%. Another Logistic Regression model was fitted with the 15 top important variables, obtained from the Variable Importance plot. This worked and the accuracy hyped to 80.95% and the Area under the curve to 0.779. Thus, we obatin the Logistic Regression with 80.95% accuracy and 77.9% AUC to classify the Attrition.

However, when compared to the best Random forest model, there is still a gap in the accuracy. Therefore, among the two models considered- Random Forest and Logistic Regression, Random forest with feature engineering done and with the optimal number of trees performed better in classifying the respinse variable- Attrition.

**Main Inferences:** (The inferences of the plots and other steps have been provided after the codes itself)

* The Random forest model with just the pre proccessed data, with the class imbalance issue solved, gave an accuracy of 82.77 %.
* The model when tuned with tunegrid and traincontrol(), resulted in a poor model performance with a decreased accuracy of 82.54%.
* So, some data wrapping was implemented to make the results better. The discretized variables were retained and the original features were removed. The model implementation with this data, improved the model performance giving an accuracy of 82.99%. The same model was checked with the parameter ntree.
* With the additon of the optimal number of trees (550) in the model, the accuracy further increased to 83.22%. Thus, the best random forest model has been obtained with an accuracy of 83.22% and AUC score of 0.676 (which is the highest among the AUCs). The Time Complexity of the different model building were same, except that the tuning of the model took more time.(noted in the results provided).
* The logistic regression model with the pre proccessed was checked for multicolinearity. The variable ‘ÝearsAtCompany’, which had VIF of above 5 was removed. The model was further fitted which gave an accuracy of 79.59%.
* In attempt of increasing the accuracy and AUC, only significant variables (10 % LOS) were considered as predictors. But, the result proved otherwise. The accuracy was decreased to 78.68%. Another Logistic Regression model was fitted with the 15 top important variables, obtained from the Variable Importance plot. This worked and the accuracy hyped to 80.95% and the Area under the curve to 0.779. Thus, we obatin the Logistic Regression with 80.95% accuracy and 77.9% AUC to classify the Attrition.
* However, when compared to the best Random forest model, there is still a gap in the accuracy. Therefore, among the two models considered- Random Forest and Logistic Regression, Random forest with feature engineering done and with the optimal number of trees performed better in classifying the response variable- Attrition.

References:

[1] Nitesh V Chawla, "Data mining for imbalanced datasets: An overview", Data mining and knowledge discovery handbook, pp. 853-867, 2005.

[2] Andrew Estabrooks, Jo Taeho and Nathalie Japkowicz, "A multiple resampling method for learning from imbalanced data sets", Computational Intelligence, vol. 20, no. 1, pp. 18-36, 2004.

[3] David J Hand and Robert J Till, "A simple generalisation of the area under the roc curve for multiple class classification problems", Machine Learning, vol. 45, no. 2, pp. 171-186, 2001.

[4] Andy Liaw and Matthew Wiener, "Classification and regression by randomforest", R News, vol. 2, no. 3, pp. 18-22, 2002.