hr2\_miniproject.R

sumedh

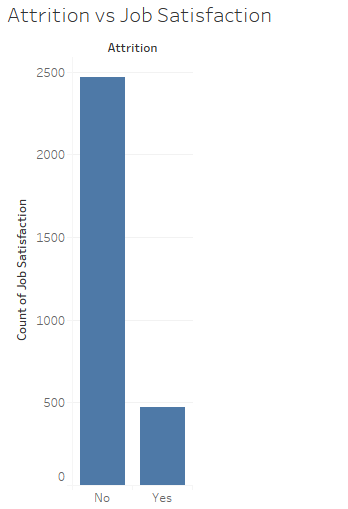
Sun Jul 08 16:49:42 2018

rm(list=ls())  
setwd("D:/Great Lakes PGPDSE/Great Lakes/13 Ensemble Techniques/Mini Project")  
hr=read.csv("hr\_working.csv",stringsAsFactors = TRUE)

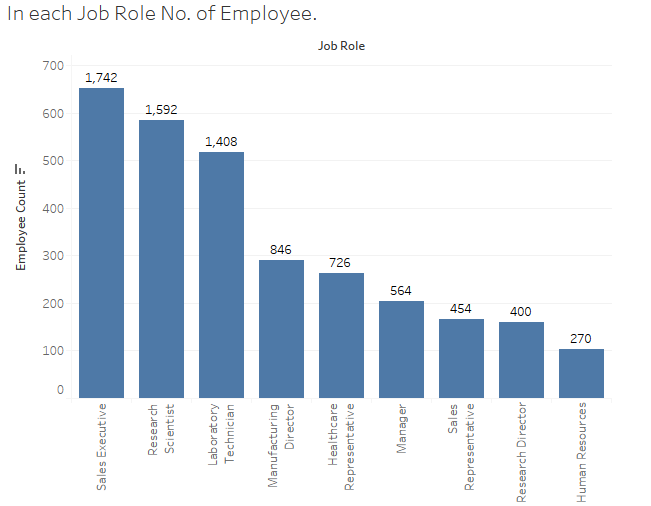
#There is no missing value in the data set

str(hr)

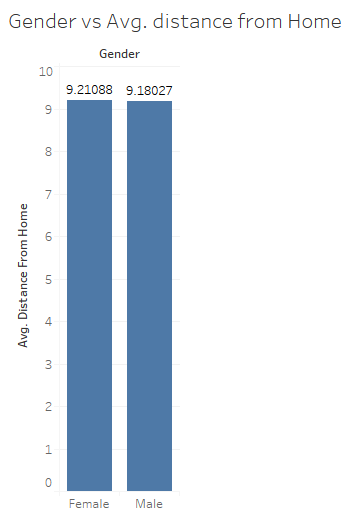
summary(hr)



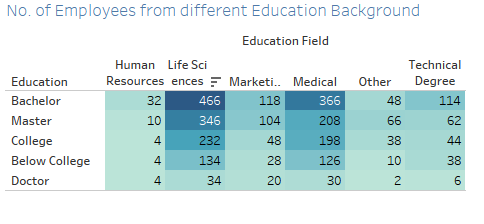
From the above graph it is clear that very less employee attrition is there due to the job satisfaction. It means that company is very good.



From the data it is clear that there are large number of employees in Sales Executives followed by research scientist. From this graph it is clear that data is from Pharmaceutical Manufacturing.

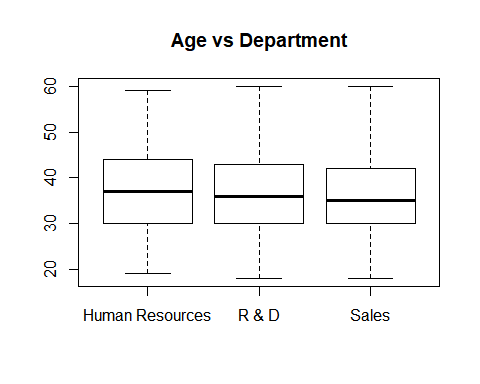


Average distance from Female and Male is almost same from the office.

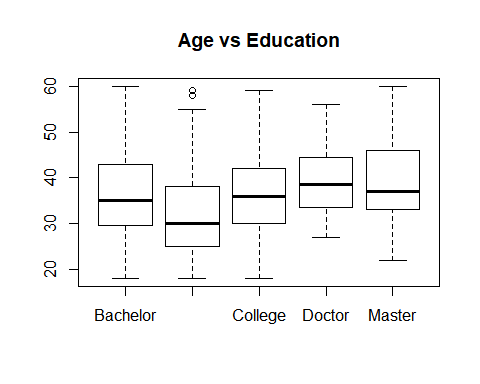


In every department maximum degree in education field were Bachelor’s degree.

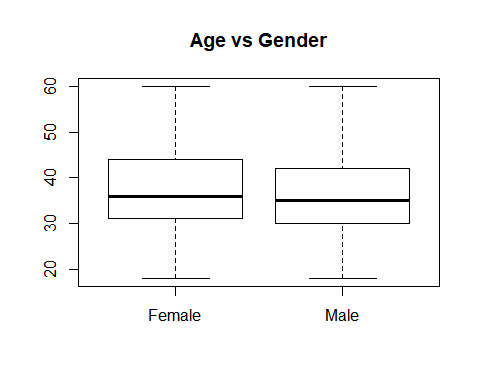
boxplot(Age~Department,data=hr,main= "Age vs Department")



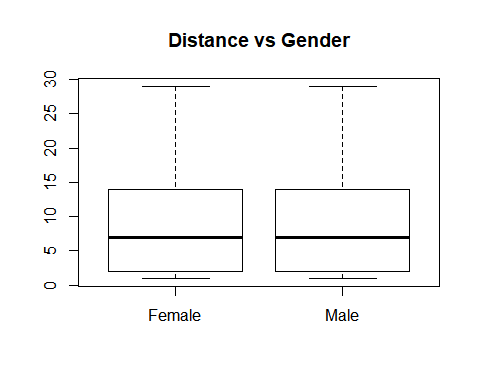
# There is no outlier in the data when comparing Age with Department.  
# Median Age of HR and R&D is appoximately same and Sales have lowest median age among all.  
boxplot(Age~Education,data=hr,main= "Age vs Education")



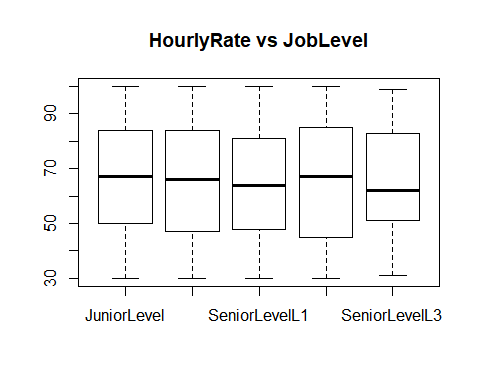
#From the above graph it is clear that Below collage have the lowest median age.  
#Bachelor and Collage have the approximately same median Age.  
#Doctor is having the highest median age.  
#Master median age is slightly above the Bachelor and collage median age.  
boxplot(Age~Gender,data=hr,main= "Age vs Gender")



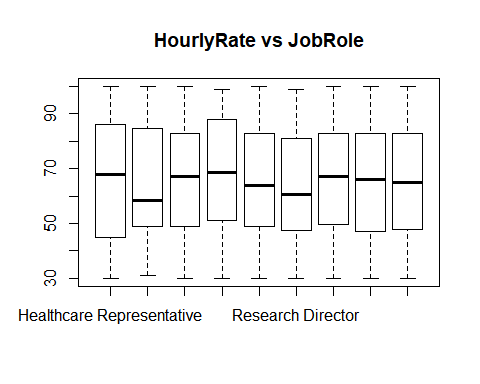
#Median Age of Female is more compare to the Male.  
boxplot(DistanceFromHome~Gender,data=hr,main= "Distance vs Gender")



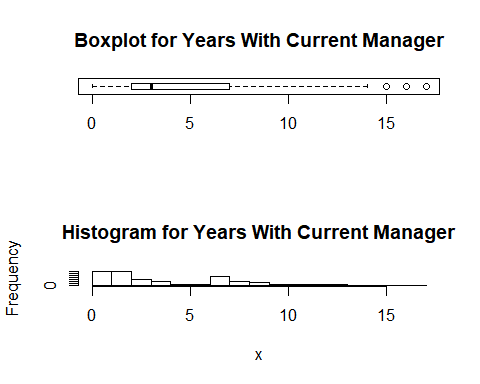
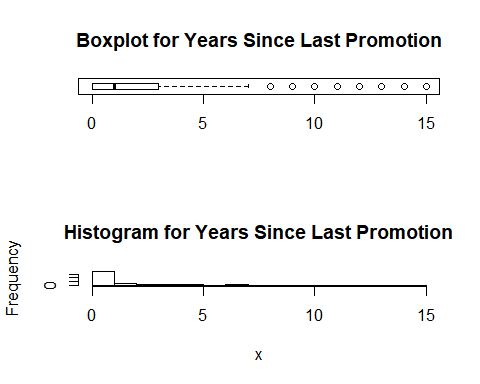
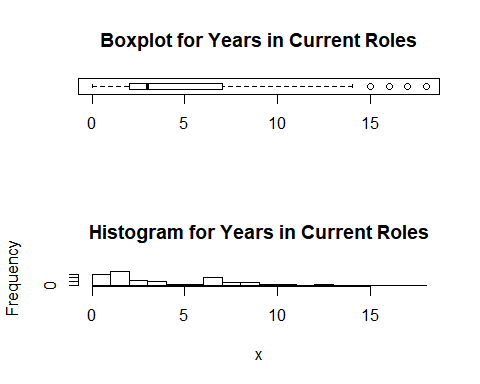
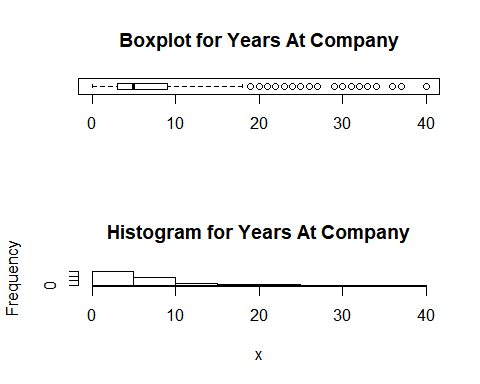
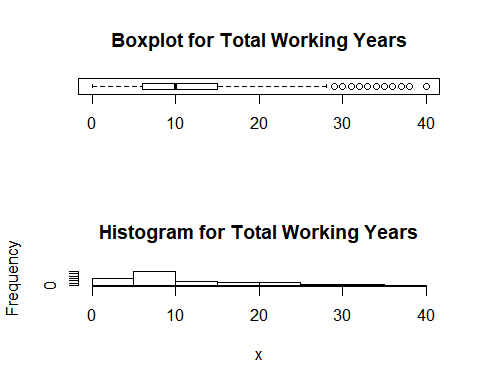
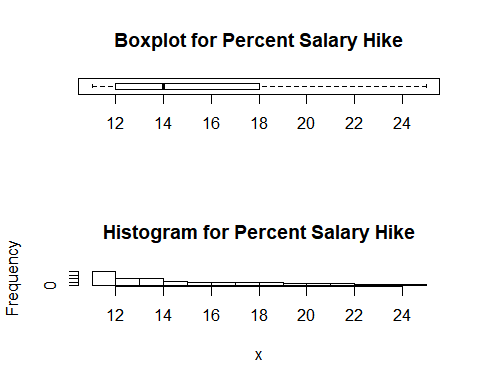
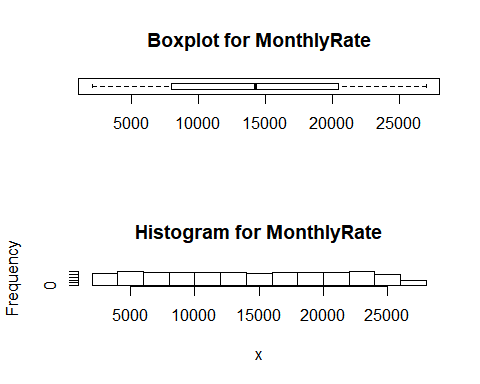
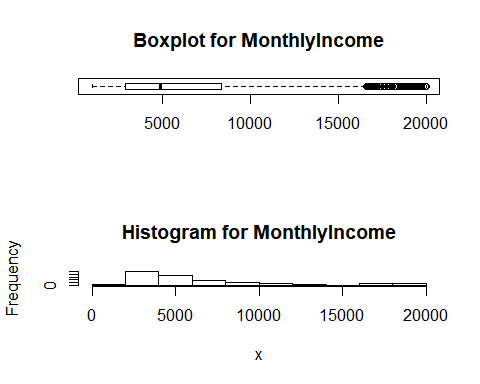
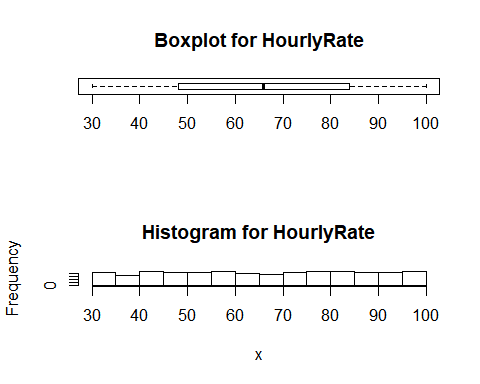
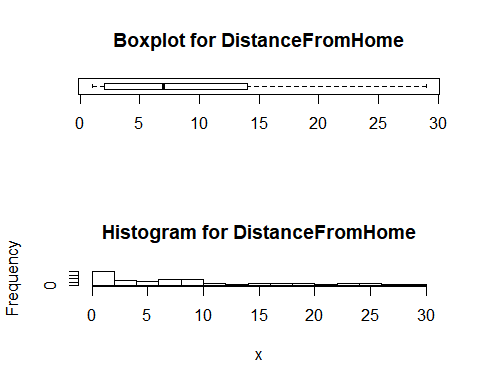
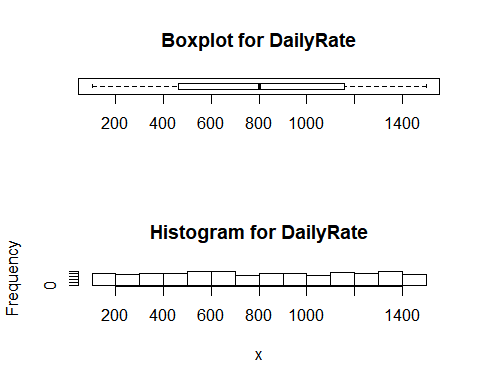
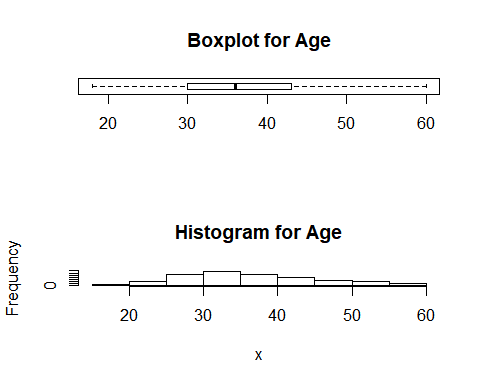
# Male and Female both median distance from home is almost same from the working location.  
# So distance from home to office does not effect much to Gender.  
boxplot(HourlyRate~JobLevel,data=hr,main= "HourlyRate vs JobLevel")



#There is a median of SeniorLevelL2 is higher hourly rate among all job level.  
boxplot(HourlyRate~JobRole,data=hr,main= "HourlyRate vs JobRole")



d=c(2,4,6,11,17,18,21,25,28,29,30,31)  
colname=c("Age","DailyRate","DistanceFromHome","HourlyRate","MonthlyIncome","MonthlyRate","Percent Salary Hike","Total Working Years","Years At Company","Years in Current Roles","Years Since Last Promotion","Years With Current Manager")  
  
hrf=hr[,c(2,4,6,11,17,18,21,25,28,29,30,31)]  
pl = colnames(hrf)  
colname=c("Age","DailyRate","DistanceFromHome","HourlyRate","MonthlyIncome","MonthlyRate","Percent Salary Hike","Total Working Years","Years At Company","Years in Current Roles","Years Since Last Promotion","Years With Current Manager")  
par(mfrow=c(2,1))  
for (i in 1:length(pl)) {  
   
 x <- hrf[,i]  
 boxplot(x,horizontal = T,main = paste("Boxplot for", colname[i]))  
 hist(x,main = paste("Histogram for", colname[i]))  
}



# From the Box plot and the Histogram for many attributes some is not showing outliers and some is showing outliers due to do the mistach between the no. of experiece , Salary , position and many other factors are also associated with outliers which is valid .  
# So we can't remove the outliers from the dataset as it hold true.

# Hourly rate of mangaer is very high among all other job role.  
  
#Target column is "Attrition" column. Convert Yes / No values to 1 / 0  
levels(hr$Attrition) <- c(0,1)  
hr$Attrition

hr$Attrition <- as.numeric(as.character(hr$Attrition))  
str(hr)

## 'data.frame': 2940 obs. of 34 variables:  
## $ Attrition : num 1 0 1 0 0 0 0 0 0 0 ...  
## $ Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 3 2 3 2 3 2 3 3 2 3 ...  
## $ DailyRate : int 1102 279 1373 1392 591 1005 1324 1358 216 1299 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...  
## $ DistanceFromHome : int 1 8 2 3 2 2 3 24 23 27 ...  
## $ Education : Factor w/ 5 levels "Bachelor","Below College",..: 3 2 3 5 2 3 1 2 1 1 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...  
## $ EmployeeCount : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ EmployeeNumber : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ EnvironmentSatisfaction : Factor w/ 4 levels "High","Low","Medium",..: 3 1 4 4 2 4 1 4 4 1 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...  
## $ HourlyRate : int 94 61 92 56 40 79 81 67 44 94 ...  
## $ JobInvolvement : Factor w/ 4 levels "High","Low","Medium",..: 1 3 3 1 1 1 4 1 3 1 ...  
## $ JobLevel : Factor w/ 5 levels "JuniorLevel",..: 2 2 1 1 1 1 1 1 3 2 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1 ...  
## $ JobSatisfaction : Factor w/ 4 levels "High","Low","Medium",..: 4 3 1 1 3 4 2 1 1 1 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...  
## $ MonthlyIncome : int 5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...  
## $ MonthlyRate : int 19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...  
## $ NumCompaniesWorked : int 8 1 6 1 9 0 4 1 0 6 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 1 1 ...  
## $ PercentSalaryHike : int 11 23 15 11 12 13 20 22 21 13 ...  
## $ PerformanceRating : Factor w/ 2 levels "Excellent","Outstanding": 1 2 1 1 1 1 2 2 2 1 ...  
## $ RelationshipSatisfaction: Factor w/ 4 levels "High","Low","Medium",..: 2 4 3 1 4 1 2 3 3 3 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : Factor w/ 4 levels "<100ESOPShares",..: 4 1 4 4 1 4 2 1 4 3 ...  
## $ TotalWorkingYears : int 8 10 7 8 6 8 12 1 10 17 ...  
## $ TrainingTimesLastYear : int 0 3 3 3 3 2 3 2 2 3 ...  
## $ WorkLifeBalance : Factor w/ 4 levels "Bad","Best","Better",..: 1 3 3 3 3 4 4 3 3 4 ...  
## $ YearsAtCompany : int 6 10 0 8 2 7 1 1 9 7 ...  
## $ YearsInCurrentRole : int 4 7 0 7 2 7 0 0 7 7 ...  
## $ YearsSinceLastPromotion : int 0 1 0 3 2 3 0 0 1 7 ...  
## $ YearsWithCurrManager : int 5 7 0 0 2 6 0 0 8 7 ...

#Split file in 70 : 30  
  
dt = sort(sample(nrow(hr), nrow(hr)\*.7))  
train<-hr[dt,]  
test<-hr[-dt,]  
  
c(nrow(train), nrow(test))

## [1] 2058 882

#Building the model using Random Forest  
  
library(randomForest)

## randomForest 4.6-14

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

?randomForest

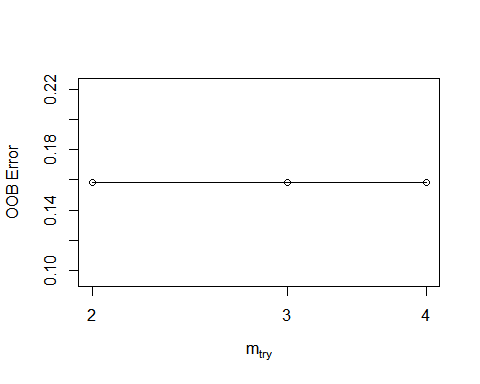
## starting httpd help server ...

## done

View(train)  
  
RF <- randomForest(as.factor(Attrition) ~ ., data = train,   
 ntree=501, mtry = 3, nodesize = 10,  
 importance=TRUE)  
  
# From the graph of Error Rates after 80 tree Error became constant  
RF$err.rate

## List the importance of the variables.  
impVar <- round(randomForest::importance(RF), 2)  
View(impVar[order(impVar[,4], decreasing=TRUE),])  
# Important List of variables   
# Overtime, Monthly Income,Job Role, Age  
  
?tuneRF  
## Tuning Random Forest  
tRF <- tuneRF(x = train[,-c(1)],   
 y=as.factor(train$Attrition),  
 mtryStart = 3,   
 ntreeTry=101,   
 stepFactor = 1.5,   
 improve = 0.0001,   
 trace=TRUE,   
 plot = TRUE,  
 doBest = TRUE,  
 nodesize = 300,   
 importance=TRUE  
)

## mtry = 3 OOB error = 15.84%   
## Searching left ...  
## mtry = 2 OOB error = 15.84%   
## 0 1e-04   
## Searching right ...  
## mtry = 4 OOB error = 15.84%   
## 0 1e-04



tRF

##   
## Call:  
## randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1], nodesize = 300, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 15.84%  
## Confusion matrix:  
## 0 1 class.error  
## 0 1732 0 0  
## 1 326 0 1

tRF$importance

## 0 1 MeanDecreaseAccuracy  
## Age 7.799908e-04 2.843172e-03 1.117089e-03  
## BusinessTravel 3.670284e-05 1.564264e-04 4.948179e-05  
## DailyRate 7.883036e-05 3.393875e-04 1.194973e-04  
## Department 3.439101e-04 -1.055870e-04 2.823921e-04  
## DistanceFromHome -3.145179e-05 1.132017e-04 -7.997584e-06  
## Education 3.444704e-05 -4.796037e-05 2.104236e-05  
## EducationField 1.962333e-04 4.016241e-06 1.629229e-04  
## EmployeeCount 0.000000e+00 0.000000e+00 0.000000e+00  
## EmployeeNumber 1.932906e-05 -2.684294e-06 1.624716e-05  
## EnvironmentSatisfaction 6.288014e-05 5.314726e-05 5.995574e-05  
## Gender -2.883946e-05 4.725218e-05 -1.600000e-05  
## HourlyRate 2.468664e-07 1.025199e-04 1.609306e-05  
## JobInvolvement 9.557349e-05 4.722506e-04 1.521886e-04  
## JobLevel 6.704022e-04 4.191511e-04 6.326958e-04  
## JobRole 6.266754e-04 8.548461e-04 6.773735e-04  
## JobSatisfaction 1.126822e-04 2.908892e-05 9.662585e-05  
## MaritalStatus 1.188784e-04 2.072841e-03 4.237177e-04  
## MonthlyIncome 3.010277e-04 1.450380e-03 4.798907e-04  
## MonthlyRate 9.055526e-05 -3.730494e-05 6.799826e-05  
## NumCompaniesWorked 3.863603e-05 -8.404771e-05 1.776191e-05  
## OverTime 9.771875e-04 4.067330e-03 1.458225e-03  
## PercentSalaryHike 6.504925e-05 5.080825e-05 6.285538e-05  
## PerformanceRating 2.471656e-05 8.299756e-05 3.404870e-05  
## RelationshipSatisfaction 5.960272e-06 -4.802921e-05 -3.109120e-06  
## StandardHours 0.000000e+00 0.000000e+00 0.000000e+00  
## StockOptionLevel 9.527702e-05 1.255739e-03 2.675961e-04  
## TotalWorkingYears 6.169893e-04 1.087154e-03 6.918750e-04  
## TrainingTimesLastYear -6.603938e-06 7.610748e-05 5.005314e-06  
## WorkLifeBalance 4.081982e-05 -2.377066e-05 3.155190e-05  
## YearsAtCompany 7.614991e-05 1.139257e-03 2.373560e-04  
## YearsInCurrentRole 3.874129e-06 1.777424e-04 2.936057e-05  
## YearsSinceLastPromotion 1.295913e-04 -1.929250e-04 8.170200e-05  
## YearsWithCurrManager 5.187925e-05 -2.501373e-04 7.351530e-06  
## MeanDecreaseGini  
## Age 3.78987547  
## BusinessTravel 1.22174612  
## DailyRate 1.14178839  
## Department 0.83695905  
## DistanceFromHome 0.62605957  
## Education 0.29046872  
## EducationField 1.36337370  
## EmployeeCount 0.00000000  
## EmployeeNumber 0.58960598  
## EnvironmentSatisfaction 1.33367789  
## Gender 0.11571973  
## HourlyRate 0.42310976  
## JobInvolvement 2.14994342  
## JobLevel 3.16824986  
## JobRole 3.51080494  
## JobSatisfaction 0.81195995  
## MaritalStatus 3.49396803  
## MonthlyIncome 3.54342575  
## MonthlyRate 0.85148512  
## NumCompaniesWorked 0.66609197  
## OverTime 6.43675841  
## PercentSalaryHike 0.54904998  
## PerformanceRating 0.07094202  
## RelationshipSatisfaction 0.74632487  
## StandardHours 0.00000000  
## StockOptionLevel 3.12975344  
## TotalWorkingYears 3.58238759  
## TrainingTimesLastYear 0.37205089  
## WorkLifeBalance 1.37654742  
## YearsAtCompany 3.26096301  
## YearsInCurrentRole 1.55373172  
## YearsSinceLastPromotion 0.51959020  
## YearsWithCurrManager 2.67768875

## List the importance of the variables.  
impVart <- round(randomForest::importance(tRF), 2)  
View(impVart[order(impVart[,4], decreasing=TRUE),])  
# After tuning the modelthe list of Important variable is  
# OverTime,YearsAtCompany,MonthlyIncome,TotalWorkingYears  
  
View(train)  
## Scoring syntax  
train$predict.class <- predict(tRF, train, type="class")  
train$predict.score <- predict(tRF, train, type="prob")  
head(train)

## Attrition Age BusinessTravel DailyRate Department DistanceFromHome  
## 1 1 41 Travel\_Rarely 1102 Sales 1  
## 2 0 49 Travel\_Frequently 279 R & D 8  
## 3 1 37 Travel\_Rarely 1373 R & D 2  
## 4 0 33 Travel\_Frequently 1392 R & D 3  
## 5 0 27 Travel\_Rarely 591 R & D 2  
## 7 0 59 Travel\_Rarely 1324 R & D 3  
## Education EducationField EmployeeCount EmployeeNumber  
## 1 College Life Sciences 1 1  
## 2 Below College Life Sciences 1 2  
## 3 College Other 1 3  
## 4 Master Life Sciences 1 4  
## 5 Below College Medical 1 5  
## 7 Bachelor Medical 1 7  
## EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel  
## 1 Medium Female 94 High MidLevel  
## 2 High Male 61 Medium MidLevel  
## 3 Very High Male 92 Medium JuniorLevel  
## 4 Very High Female 56 High JuniorLevel  
## 5 Low Male 40 High JuniorLevel  
## 7 High Female 81 Very High JuniorLevel  
## JobRole JobSatisfaction MaritalStatus MonthlyIncome  
## 1 Sales Executive Very High Single 5993  
## 2 Research Scientist Medium Married 5130  
## 3 Laboratory Technician High Single 2090  
## 4 Research Scientist High Married 2909  
## 5 Laboratory Technician Medium Married 3468  
## 7 Laboratory Technician Low Married 2670  
## MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike  
## 1 19479 8 Yes 11  
## 2 24907 1 No 23  
## 3 2396 6 Yes 15  
## 4 23159 1 Yes 11  
## 5 16632 9 No 12  
## 7 9964 4 Yes 20  
## PerformanceRating RelationshipSatisfaction StandardHours  
## 1 Excellent Low 80  
## 2 Outstanding Very High 80  
## 3 Excellent Medium 80  
## 4 Excellent High 80  
## 5 Excellent Very High 80  
## 7 Outstanding Low 80  
## StockOptionLevel TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  
## 1 NoStockOption 8 0 Bad  
## 2 <100ESOPShares 10 3 Better  
## 3 NoStockOption 7 3 Better  
## 4 NoStockOption 8 3 Better  
## 5 <100ESOPShares 6 3 Better  
## 7 >500ESOPShares 12 3 Good  
## YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion  
## 1 6 4 0  
## 2 10 7 1  
## 3 0 0 0  
## 4 8 7 3  
## 5 2 2 2  
## 7 1 0 0  
## YearsWithCurrManager predict.class predict.score.0 predict.score.1  
## 1 5 0 0.932 0.068  
## 2 7 0 0.982 0.018  
## 3 0 0 0.930 0.070  
## 4 0 0 0.974 0.026  
## 5 2 0 0.992 0.008  
## 7 0 0 0.980 0.020

class(train$predict.score)

## [1] "matrix" "votes"

## deciling  
## deciling code  
decile <- function(x){  
 deciles <- vector(length=10)  
 for (i in seq(0.1,1,.1)){  
 deciles[i\*10] <- quantile(x, i, na.rm=T)  
 }  
 return (  
 ifelse(x<deciles[1], 1,  
 ifelse(x<deciles[2], 2,  
 ifelse(x<deciles[3], 3,  
 ifelse(x<deciles[4], 4,  
 ifelse(x<deciles[5], 5,  
 ifelse(x<deciles[6], 6,  
 ifelse(x<deciles[7], 7,  
 ifelse(x<deciles[8], 8,  
 ifelse(x<deciles[9], 9, 10  
 ))))))))))  
}  
  
  
train$deciles <- decile(train$predict.score[,2])  
  
  
library(data.table)  
tmp\_DT = data.table(train)  
rank <- tmp\_DT[, list(  
 cnt = length(Attrition),   
 cnt\_resp = sum(Attrition),   
 cnt\_non\_resp = sum(Attrition == 0)) ,   
 by=deciles][order(-deciles)]  
rank$rrate <- round (rank$cnt\_resp / rank$cnt,2);  
rank$cum\_resp <- cumsum(rank$cnt\_resp)  
rank$cum\_non\_resp <- cumsum(rank$cnt\_non\_resp)  
rank$cum\_rel\_resp <- round(rank$cum\_resp / sum(rank$cnt\_resp),2);  
rank$cum\_rel\_non\_resp <- round(rank$cum\_non\_resp / sum(rank$cnt\_non\_resp),2);  
rank$ks <- abs(rank$cum\_rel\_resp - rank$cum\_rel\_non\_resp);  
  
  
library(scales)  
rank$rrate <- percent(rank$rrate)  
rank$cum\_rel\_resp <- percent(rank$cum\_rel\_resp)  
rank$cum\_rel\_non\_resp <- percent(rank$cum\_rel\_non\_resp)  
  
View(rank)  
# Ks statistics for train data is 8th decile and highest KS Statistics is 0.52  
sum(train$Attrition) / nrow(train)

## [1] 0.1584062

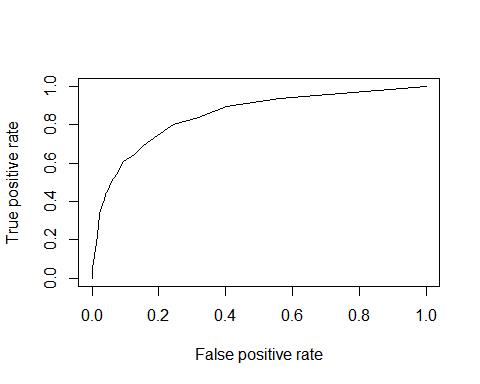
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

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

pred <- prediction(train$predict.score[,2], train$Attrition)  
perf <- performance(pred, "tpr", "fpr")  
plot(perf)



KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])  
KS ##0.5337481

## [1] 0.5611868

## Area Under Curve  
auc <- performance(pred,"auc");   
auc <- as.numeric(auc@y.values)  
auc ## 0.8288224

## [1] 0.849293

## Gini Coefficient  
library(ineq)  
gini = ineq(train$predict.score[,2], type="Gini")  
gini ##0.7692009

## [1] 0.7303286

## Classification Error  
with(train, table(Attrition, predict.class))

## predict.class  
## Attrition 0 1  
## 0 1732 0  
## 1 326 0

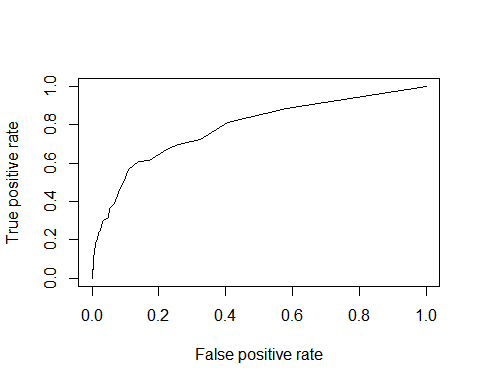
# Mis Classification Error 328/(1730+328) =15.93%  
# Accuracy for train data 84.06%  
  
## Scoring syntax  
test$predict.class <- predict(tRF, test, type="class")  
test$predict.score <- predict(tRF, test, type="prob")  
head(test)

## Attrition Age BusinessTravel DailyRate Department DistanceFromHome  
## 6 0 32 Travel\_Frequently 1005 R & D 2  
## 8 0 30 Travel\_Rarely 1358 R & D 24  
## 14 0 34 Travel\_Rarely 1346 R & D 19  
## 16 0 29 Travel\_Rarely 1389 R & D 21  
## 17 0 32 Travel\_Rarely 334 R & D 5  
## 20 0 38 Travel\_Rarely 371 R & D 2  
## Education EducationField EmployeeCount EmployeeNumber  
## 6 College Life Sciences 1 6  
## 8 Below College Life Sciences 1 8  
## 14 College Medical 1 14  
## 16 Master Life Sciences 1 16  
## 17 College Life Sciences 1 17  
## 20 Bachelor Life Sciences 1 20  
## EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel  
## 6 Very High Male 79 High JuniorLevel  
## 8 Very High Male 67 High JuniorLevel  
## 14 Medium Male 93 High JuniorLevel  
## 16 Medium Female 51 Very High SeniorLevelL1  
## 17 Low Male 80 Very High JuniorLevel  
## 20 Very High Male 45 High JuniorLevel  
## JobRole JobSatisfaction MaritalStatus MonthlyIncome  
## 6 Laboratory Technician Very High Single 3068  
## 8 Laboratory Technician High Divorced 2693  
## 14 Laboratory Technician Very High Divorced 2661  
## 16 Manufacturing Director Low Divorced 9980  
## 17 Research Scientist Medium Divorced 3298  
## 20 Research Scientist Very High Single 3944  
## MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike  
## 6 11864 0 No 13  
## 8 13335 1 No 22  
## 14 8758 0 No 11  
## 16 10195 1 No 11  
## 17 15053 0 Yes 12  
## 20 4306 5 Yes 11  
## PerformanceRating RelationshipSatisfaction StandardHours  
## 6 Excellent High 80  
## 8 Outstanding Medium 80  
## 14 Excellent High 80  
## 16 Excellent High 80  
## 17 Excellent Very High 80  
## 20 Excellent High 80  
## StockOptionLevel TotalWorkingYears TrainingTimesLastYear  
## 6 NoStockOption 8 2  
## 8 <100ESOPShares 1 2  
## 14 <100ESOPShares 3 2  
## 16 <100ESOPShares 10 1  
## 17 100-500ESOPShares 7 5  
## 20 NoStockOption 6 3  
## WorkLifeBalance YearsAtCompany YearsInCurrentRole  
## 6 Good 7 7  
## 8 Better 1 0  
## 14 Better 2 2  
## 16 Better 10 9  
## 17 Good 6 2  
## 20 Better 3 2  
## YearsSinceLastPromotion YearsWithCurrManager predict.class  
## 6 3 6 0  
## 8 0 0 0  
## 14 1 2 0  
## 16 8 8 0  
## 17 0 5 0  
## 20 1 2 0  
## predict.score.0 predict.score.1  
## 6 0.986 0.014  
## 8 0.988 0.012  
## 14 0.994 0.006  
## 16 0.996 0.004  
## 17 0.982 0.018  
## 20 0.974 0.026

class(test$predict.score)

## [1] "matrix" "votes"

test$deciles <- decile(test$predict.score[,2])  
  
tmp\_DT = data.table(test)  
h\_rank <- tmp\_DT[, list(  
 cnt = length(Attrition),   
 cnt\_resp = sum(Attrition),   
 cnt\_non\_resp = sum(Attrition == 0)) ,   
 by=deciles][order(-deciles)]  
h\_rank$rrate <- round (h\_rank$cnt\_resp / h\_rank$cnt,2);  
h\_rank$cum\_resp <- cumsum(h\_rank$cnt\_resp)  
h\_rank$cum\_non\_resp <- cumsum(h\_rank$cnt\_non\_resp)  
h\_rank$cum\_rel\_resp <- round(h\_rank$cum\_resp / sum(h\_rank$cnt\_resp),2);  
h\_rank$cum\_rel\_non\_resp <- round(h\_rank$cum\_non\_resp / sum(h\_rank$cnt\_non\_resp),2);  
h\_rank$ks <- abs(h\_rank$cum\_rel\_resp - h\_rank$cum\_rel\_non\_resp);  
  
  
library(scales)  
h\_rank$rrate <- percent(h\_rank$rrate)  
h\_rank$cum\_rel\_resp <- percent(h\_rank$cum\_rel\_resp)  
h\_rank$cum\_rel\_non\_resp <- percent(h\_rank$cum\_rel\_non\_resp)  
  
View(h\_rank)  
# Highest KS value is for 8th Decile having 46% kS Statistics  
library(ROCR)  
pred <- prediction(test$predict.score[,2], test$Attrition)  
perf <- performance(pred, "tpr", "fpr")  
plot(perf)



KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])  
KS ##0.4331515

## [1] 0.4677811

## Area Under Curve  
auc <- performance(pred,"auc");   
auc <- as.numeric(auc@y.values)  
auc ## 0.7908788

## [1] 0.785786

## Gini Coefficient  
library(ineq)  
gini = ineq(test$predict.score[,2], type="Gini")  
gini ##0.7180185

## [1] 0.7164056

## Classification Error  
with(test, table(Attrition, predict.class))

## predict.class  
## Attrition 0 1  
## 0 734 0  
## 1 148 0

# Mis Classification Error 328/(1730+328) =16.55%  
# Accuracy for test data 83.44%

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. | Model Performance Measure | Train Data | Test Data |
| 1. | Ks statistics | 0.53 | 0.46 |
| 2. | AUC | 0.83 | 0.80 |
| 3. | Gini | 0.77 | 0.72 |
| 4. | Misclassification Error | 15.93% | 16.55% |
| 5. | Accuracy | 84.06% | 83.44% |

From the model performance measure summary shows that Train and Test data are fitting correctly on the random forest model.