hr\_adaboost.R

sumedh

Sun Jul 08 20:25:24 2018

rm(list=ls())  
# Setting working directory  
setwd("D:/Great Lakes PGPDSE/Great Lakes/13 Ensemble Techniques/Mini Project")  
#Reading the Data Set in csv format  
hr=read.csv("HR\_Employee\_Attrition\_Data.csv",stringsAsFactors = TRUE,header = TRUE)  
  
View(hr)  
c(nrow(hr))

## [1] 2940

str(hr)

## 'data.frame': 2940 obs. of 34 variables:  
## $ Attrition : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...  
## $ 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 : int 2 1 2 4 1 2 3 1 3 3 ...  
## $ 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 : int 2 3 4 4 1 4 3 4 4 3 ...  
## $ 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 : int 3 2 2 3 3 3 4 3 2 3 ...  
## $ JobLevel : int 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 : int 4 2 3 3 2 4 1 3 3 3 ...  
## $ 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 : int 3 4 3 3 3 3 4 4 4 3 ...  
## $ RelationshipSatisfaction: int 1 4 2 3 4 3 1 2 2 2 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : int 0 1 0 0 1 0 3 1 0 2 ...  
## $ 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 : int 1 3 3 3 3 2 2 3 3 2 ...  
## $ 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 ...

summary(hr)

## Attrition Age BusinessTravel DailyRate   
## No :2466 Min. :18.00 Non-Travel : 300 Min. : 102.0   
## Yes: 474 1st Qu.:30.00 Travel\_Frequently: 554 1st Qu.: 465.0   
## Median :36.00 Travel\_Rarely :2086 Median : 802.0   
## Mean :36.92 Mean : 802.5   
## 3rd Qu.:43.00 3rd Qu.:1157.0   
## Max. :60.00 Max. :1499.0   
##   
## Department DistanceFromHome Education   
## Human Resources : 126 Min. : 1.000 Min. :1.000   
## Research & Development:1922 1st Qu.: 2.000 1st Qu.:2.000   
## Sales : 892 Median : 7.000 Median :3.000   
## Mean : 9.193 Mean :2.913   
## 3rd Qu.:14.000 3rd Qu.:4.000   
## Max. :29.000 Max. :5.000   
##   
## EducationField EmployeeCount EmployeeNumber   
## Human Resources : 54 Min. :1 Min. : 1.0   
## Life Sciences :1212 1st Qu.:1 1st Qu.: 735.8   
## Marketing : 318 Median :1 Median :1470.5   
## Medical : 928 Mean :1 Mean :1470.5   
## Other : 164 3rd Qu.:1 3rd Qu.:2205.2   
## Technical Degree: 264 Max. :1 Max. :2940.0   
##   
## EnvironmentSatisfaction Gender HourlyRate JobInvolvement  
## Min. :1.000 Female:1176 Min. : 30.00 Min. :1.00   
## 1st Qu.:2.000 Male :1764 1st Qu.: 48.00 1st Qu.:2.00   
## Median :3.000 Median : 66.00 Median :3.00   
## Mean :2.722 Mean : 65.89 Mean :2.73   
## 3rd Qu.:4.000 3rd Qu.: 84.00 3rd Qu.:3.00   
## Max. :4.000 Max. :100.00 Max. :4.00   
##   
## JobLevel JobRole JobSatisfaction  
## Min. :1.000 Sales Executive :652 Min. :1.000   
## 1st Qu.:1.000 Research Scientist :584 1st Qu.:2.000   
## Median :2.000 Laboratory Technician :518 Median :3.000   
## Mean :2.064 Manufacturing Director :290 Mean :2.729   
## 3rd Qu.:3.000 Healthcare Representative:262 3rd Qu.:4.000   
## Max. :5.000 Manager :204 Max. :4.000   
## (Other) :430   
## MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked  
## Divorced: 654 Min. : 1009 Min. : 2094 Min. :0.000   
## Married :1346 1st Qu.: 2911 1st Qu.: 8045 1st Qu.:1.000   
## Single : 940 Median : 4919 Median :14236 Median :2.000   
## Mean : 6503 Mean :14313 Mean :2.693   
## 3rd Qu.: 8380 3rd Qu.:20462 3rd Qu.:4.000   
## Max. :19999 Max. :26999 Max. :9.000   
##   
## OverTime PercentSalaryHike PerformanceRating RelationshipSatisfaction  
## No :2108 Min. :11.00 Min. :3.000 Min. :1.000   
## Yes: 832 1st Qu.:12.00 1st Qu.:3.000 1st Qu.:2.000   
## Median :14.00 Median :3.000 Median :3.000   
## Mean :15.21 Mean :3.154 Mean :2.712   
## 3rd Qu.:18.00 3rd Qu.:3.000 3rd Qu.:4.000   
## Max. :25.00 Max. :4.000 Max. :4.000   
##   
## StandardHours StockOptionLevel TotalWorkingYears TrainingTimesLastYear  
## Min. :80 Min. :0.0000 Min. : 0.00 Min. :0.000   
## 1st Qu.:80 1st Qu.:0.0000 1st Qu.: 6.00 1st Qu.:2.000   
## Median :80 Median :1.0000 Median :10.00 Median :3.000   
## Mean :80 Mean :0.7939 Mean :11.28 Mean :2.799   
## 3rd Qu.:80 3rd Qu.:1.0000 3rd Qu.:15.00 3rd Qu.:3.000   
## Max. :80 Max. :3.0000 Max. :40.00 Max. :6.000   
##   
## WorkLifeBalance YearsAtCompany YearsInCurrentRole  
## Min. :1.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.:2.000 1st Qu.: 3.000 1st Qu.: 2.000   
## Median :3.000 Median : 5.000 Median : 3.000   
## Mean :2.761 Mean : 7.008 Mean : 4.229   
## 3rd Qu.:3.000 3rd Qu.: 9.000 3rd Qu.: 7.000   
## Max. :4.000 Max. :40.000 Max. :18.000   
##   
## YearsSinceLastPromotion YearsWithCurrManager  
## Min. : 0.000 Min. : 0.000   
## 1st Qu.: 0.000 1st Qu.: 2.000   
## Median : 1.000 Median : 3.000   
## Mean : 2.188 Mean : 4.123   
## 3rd Qu.: 3.000 3rd Qu.: 7.000   
## Max. :15.000 Max. :17.000   
##

#Target column is "Attrition" column. Convert Yes / No values to 1 / 0  
  
levels(hr$Attrition) <- c(0,1)  
hr$Attrition

## [1] 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1  
## [35] 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [69] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0  
## [103] 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0  
## [137] 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [171] 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  
## [205] 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0  
## [239] 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1  
## [273] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0  
## [307] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0  
## [341] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0  
## [375] 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0  
## [409] 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0  
## [443] 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0  
## [477] 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0  
## [511] 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0  
## [545] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0  
## [579] 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0  
## [613] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1  
## [647] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0  
## [681] 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0  
## [715] 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0  
## [749] 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0  
## [783] 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0  
## [817] 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1  
## [851] 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0  
## [885] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0  
## [919] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0  
## [953] 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1  
## [987] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 0 0  
## [1021] 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [1055] 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0  
## [1089] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0  
## [1123] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0  
## [1157] 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0  
## [1191] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1  
## [1225] 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1  
## [1259] 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1  
## [1293] 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0  
## [1327] 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0  
## [1361] 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0  
## [1395] 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [1429] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1  
## [1463] 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0  
## [1497] 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0  
## [1531] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [1565] 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1  
## [1599] 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [1633] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0  
## [1667] 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1  
## [1701] 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [1735] 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0  
## [1769] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0  
## [1803] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0  
## [1837] 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [1871] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0  
## [1905] 0 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0  
## [1939] 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0  
## [1973] 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0  
## [2007] 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0  
## [2041] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0  
## [2075] 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0  
## [2109] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 0 0  
## [2143] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0  
## [2177] 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0  
## [2211] 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0  
## [2245] 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0  
## [2279] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0  
## [2313] 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0  
## [2347] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [2381] 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0  
## [2415] 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0  
## [2449] 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0  
## [2483] 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 0  
## [2517] 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0  
## [2551] 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0  
## [2585] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  
## [2619] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [2653] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0  
## [2687] 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1  
## [2721] 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0  
## [2755] 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0  
## [2789] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0  
## [2823] 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0  
## [2857] 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [2891] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0  
## [2925] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0  
## Levels: 0 1

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 : int 2 1 2 4 1 2 3 1 3 3 ...  
## $ 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 : int 2 3 4 4 1 4 3 4 4 3 ...  
## $ 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 : int 3 2 2 3 3 3 4 3 2 3 ...  
## $ JobLevel : int 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 : int 4 2 3 3 2 4 1 3 3 3 ...  
## $ 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 : int 3 4 3 3 3 3 4 4 4 3 ...  
## $ RelationshipSatisfaction: int 1 4 2 3 4 3 1 2 2 2 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : int 0 1 0 0 1 0 3 1 0 2 ...  
## $ 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 : int 1 3 3 3 3 2 2 3 3 2 ...  
## $ 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 data set into 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

#Build AdaBoosting Model  
  
  
library(gbm)

## Loading required package: survival

## Loading required package: lattice

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3

?gbm

## starting httpd help server ...

## done

##bernoulli means binomial data type  
set.seed(1212)  
gbmFit <- gbm(  
 formula = Attrition~.,  
 distribution = "adaboost",  
 data = train,  
 n.trees = 50, # number of trees  
 cv.folds = 10, # do 10-fold cross-validation  
 shrinkage = 1, # shrinkage or learning rate  
 bag.fraction = 1 # no dataleft aside  
)

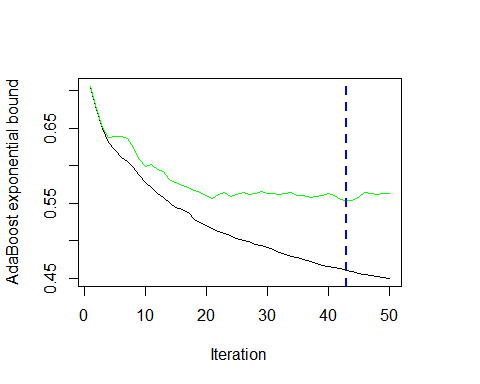
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.

## Print the gbm Fit summary  
print(gbmFit)

## gbm(formula = Attrition ~ ., distribution = "adaboost", data = train,   
## n.trees = 50, shrinkage = 1, bag.fraction = 1, cv.folds = 10)  
## A gradient boosted model with adaboost loss function.  
## 50 iterations were performed.  
## The best cross-validation iteration was 43.  
## There were 33 predictors of which 24 had non-zero influence.

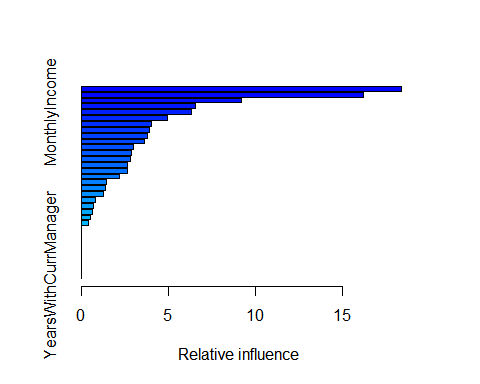
# Output from the Model   
# A gradient boosted model with adaboost loss function.  
# 50 iterations were performed.  
# The best cross-validation iteration was 44.  
# There were 33 predictors of which 25 had non-zero influence.  
  
?summary  
# Get the Best No. of Iterations using cross-validation  
best.iter <- gbm.perf(gbmFit,method="cv")



print(best.iter)

## [1] 43

summary(gbmFit, n.trees = best.iter)



## var rel.inf  
## OverTime OverTime 18.4106035  
## TotalWorkingYears TotalWorkingYears 16.2371294  
## StockOptionLevel StockOptionLevel 9.2035140  
## JobRole JobRole 6.5492253  
## MonthlyIncome MonthlyIncome 6.3451918  
## YearsSinceLastPromotion YearsSinceLastPromotion 4.9297691  
## NumCompaniesWorked NumCompaniesWorked 4.0072112  
## EnvironmentSatisfaction EnvironmentSatisfaction 3.9168012  
## RelationshipSatisfaction RelationshipSatisfaction 3.8045317  
## YearsAtCompany YearsAtCompany 3.5827498  
## DistanceFromHome DistanceFromHome 2.9948056  
## JobSatisfaction JobSatisfaction 2.8330341  
## BusinessTravel BusinessTravel 2.8071319  
## DailyRate DailyRate 2.6544005  
## EducationField EducationField 2.6110906  
## Age Age 2.1507825  
## JobInvolvement JobInvolvement 1.3989168  
## TrainingTimesLastYear TrainingTimesLastYear 1.3762396  
## WorkLifeBalance WorkLifeBalance 1.2382793  
## EmployeeNumber EmployeeNumber 0.7852750  
## Gender Gender 0.6759715  
## PercentSalaryHike PercentSalaryHike 0.5956275  
## MaritalStatus MaritalStatus 0.5115443  
## YearsInCurrentRole YearsInCurrentRole 0.3801737  
## Department Department 0.0000000  
## Education Education 0.0000000  
## EmployeeCount EmployeeCount 0.0000000  
## HourlyRate HourlyRate 0.0000000  
## JobLevel JobLevel 0.0000000  
## MonthlyRate MonthlyRate 0.0000000  
## PerformanceRating PerformanceRating 0.0000000  
## StandardHours StandardHours 0.0000000  
## YearsWithCurrManager YearsWithCurrManager 0.0000000

# Maximum relative value are in following Order.  
# TotalWorkingYears,OverTime,JobRole,Stock Option Level  
  
# predict on the new data (training data set) using "best" number of trees  
train$predict.score <- predict(   
 gbmFit, newdata=train,  
 n.trees = best.iter,type = "response")  
  
View(train)  
class(train$predict.score)

## [1] "numeric"

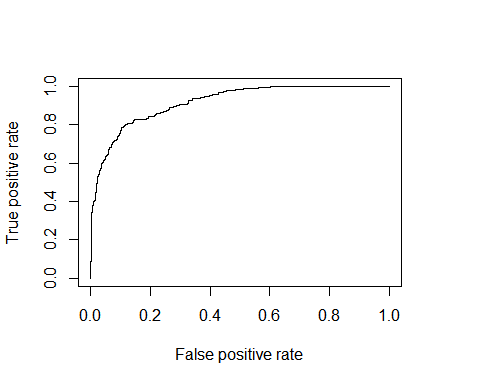
## deciling code  
decile <- function(x){  
 deciles <- vector(length=10)  
 for (i in seq(0.1,1,.1)){  
 deciles[i\*10] <- quantile(x, i, na.rm=T)  
 }  
 return (  
 ifelse(x<deciles[1], 1,  
 ifelse(x<deciles[2], 2,  
 ifelse(x<deciles[3], 3,  
 ifelse(x<deciles[4], 4,  
 ifelse(x<deciles[5], 5,  
 ifelse(x<deciles[6], 6,  
 ifelse(x<deciles[7], 7,  
 ifelse(x<deciles[8], 8,  
 ifelse(x<deciles[9], 9, 10  
 ))))))))))  
}  
  
  
## deciling  
train$deciles <- decile(train$predict.score)  
  
## Ranking code  
##install.packages("data.table")  
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 \* 100 / rank$cnt,2);  
rank$cum\_resp <- cumsum(rank$cnt\_resp)  
rank$cum\_non\_resp <- cumsum(rank$cnt\_non\_resp)  
rank$cum\_perct\_resp <- round(rank$cum\_resp \* 100 / sum(rank$cnt\_resp),2);  
rank$cum\_perct\_non\_resp <- round(rank$cum\_non\_resp \* 100 / sum(rank$cnt\_non\_resp),2);  
rank$ks <- abs(rank$cum\_perct\_resp - rank$cum\_perct\_non\_resp);  
  
View(rank)  
  
  
  
##install.packages("ROCR")  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

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

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



KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])  
auc <- performance(pred,"auc");   
auc <- as.numeric(auc@y.values)  
  
#install.packages("MLmetrics")  
library(MLmetrics)

## Warning: package 'MLmetrics' was built under R version 3.5.1

##   
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':  
##   
## Recall

gini = Gini(train$predict.score, train$Attrition)  
  
auc ## 0.9168812

## [1] 0.9170924

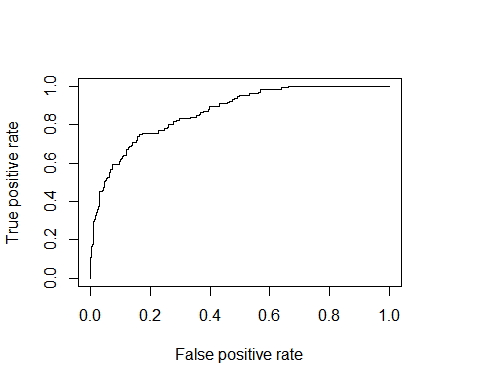
KS ##0.6841633

## [1] 0.6851312

gini ##0.8337624

## [1] 0.8341847

##concordance is same as auc only approach is different  
  
#### Testing on test dataset   
  
test$predict.score <-   
 predict( gbmFit,newdata=test,  
 n.trees = best.iter, type = "response")  
test$deciles <- decile(test$predict.score)  
  
## Ranking code  
library(data.table)  
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 \* 100 / 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\_perct\_resp <- round(h\_rank$cum\_resp \* 100 / sum(h\_rank$cnt\_resp),2);  
h\_rank$cum\_perct\_non\_resp <- round(h\_rank$cum\_non\_resp \* 100 / sum(h\_rank$cnt\_non\_resp),2);  
h\_rank$ks <- abs(h\_rank$cum\_perct\_resp - h\_rank$cum\_perct\_non\_resp);  
  
View(h\_rank)  
  
  
  
pred <- prediction(test$predict.score, test$Attrition)  
perf <- performance(pred, "tpr", "fpr")  
plot(perf)



KS\_h <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])  
auc\_h <- performance(pred,"auc");   
auc\_h <- as.numeric(auc\_h@y.values)  
gini\_h = Gini(test$predict.score, test$Attrition)  
  
auc\_h #0.8637535

## [1] 0.8676675

KS\_h #0.5693627

## [1] 0.5856415

gini\_h #0.727507

## [1] 0.7353351

auc #0.9168812

## [1] 0.9170924

KS #0.6841633

## [1] 0.6851312

gini #0.8337624

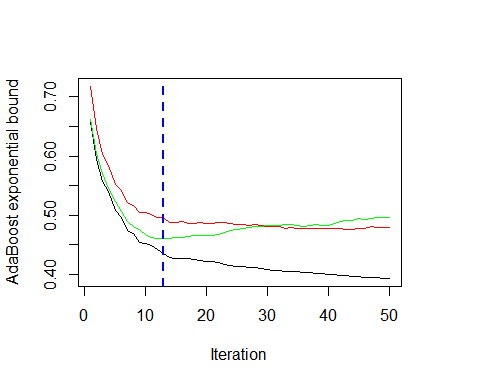
## [1] 0.8341847

**## We begin the GBM Model Tuning in this section ##  
  
## Import the Data again beacuse in the previous steps predict.score column got added**  
  
## Create Grid Search Dataframe  
?expand.grid  
?expand.grid  
hyper\_grid <- expand.grid(  
 shrinkage = c(0.3, 1), ## You may try more shrinkage parameters  
 interaction.depth = c(1, 3), ## You may try more interaction depth  
 bag.fraction = c(.8, 1), ## You may try with different values  
 n.minobsinnode = c(10), ## You may try different Min Obs combinations  
 optimal\_trees = NA, # a parameter to capture iteration results  
 valid\_error = NA # a parameter to capture iteration results  
)  
  
View(hyper\_grid)  
library(gbm)   
set.seed(1212) ## Set the seed to ensure reproducibility  
start\_time<-proc.time() ## Start Process Time  
for(i in 1:nrow(hyper\_grid)) { ## grid search  
 gbm.tune <- gbm(  
 formula = Attrition~.,  
 distribution = "adaboost",  
 data =train,  
 n.trees = 50,  
 interaction.depth = hyper\_grid$interaction.depth[i],  
 shrinkage = hyper\_grid$shrinkage[i],  
 n.minobsinnode = hyper\_grid$n.minobsinnode[i],  
 bag.fraction = hyper\_grid$bag.fraction[i],  
 train.fraction = .75,  
 cv.folds = 10  
 )  
 best.iter <- gbm.perf(gbm.tune,method="cv")  
 hyper\_grid$optimal\_trees[i] <- best.iter   
 hyper\_grid$valid\_error[i] <- gbm.tune$valid.error[best.iter]  
}

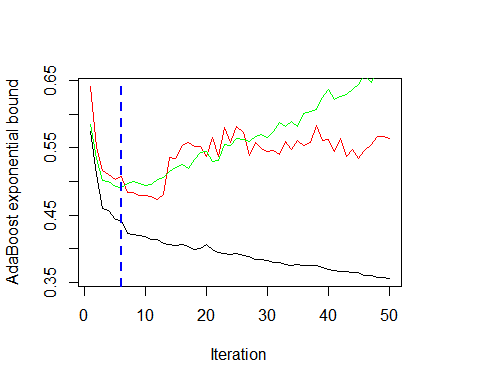
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.

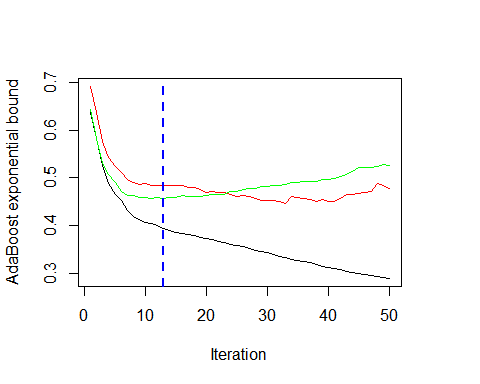
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



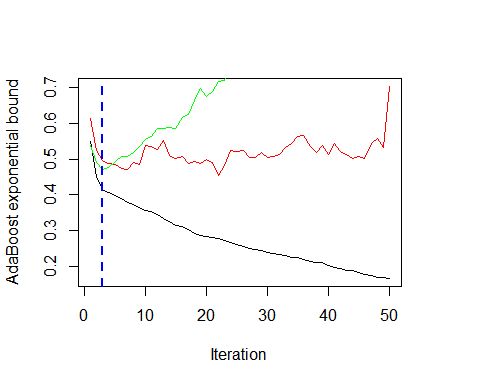
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



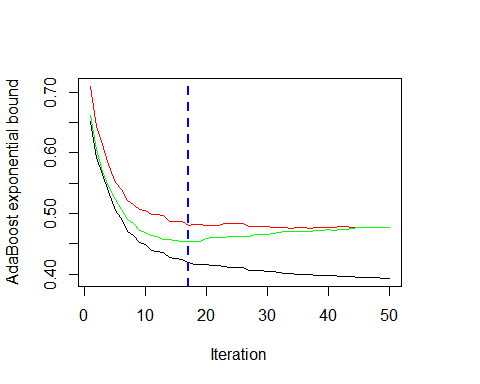
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



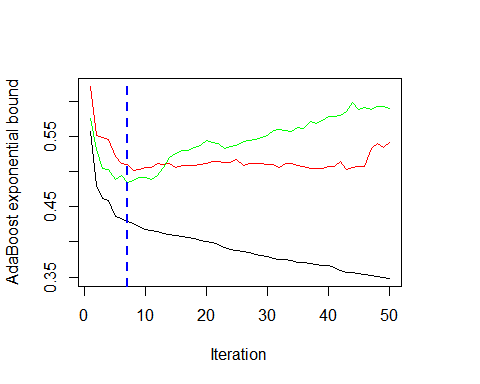
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



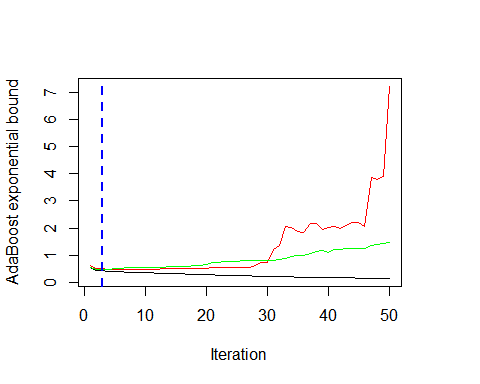
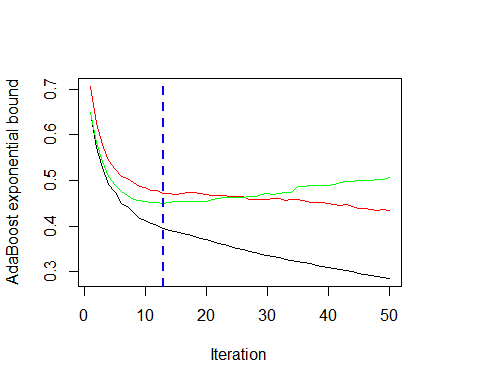
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 8: EmployeeCount has no variation.  
  
## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =  
## w, : variable 25: StandardHours has no variation.



end\_time<-proc.time() ## End Process Time  
proc\_time=end\_time - start\_time  
proc\_time ## Total Process Time

## user system elapsed   
## 6.33 2.30 149.39

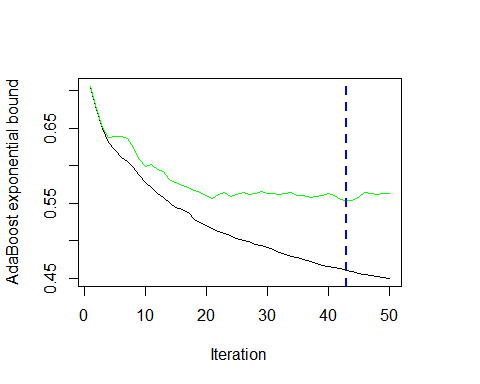
'proc\_time ## Total Process Time  
 user system elapsed   
6.64 2.44 163.22'

## [1] "proc\_time ## Total Process Time\n user system elapsed \n6.64 2.44 163.22"

head(hyper\_grid)

## shrinkage interaction.depth bag.fraction n.minobsinnode optimal\_trees  
## 1 0.3 1 0.8 10 13  
## 2 1.0 1 0.8 10 6  
## 3 0.3 3 0.8 10 13  
## 4 1.0 3 0.8 10 3  
## 5 0.3 1 1.0 10 17  
## 6 1.0 1 1.0 10 7  
## valid\_error  
## 1 0.4968809  
## 2 0.5071009  
## 3 0.4834009  
## 4 0.4965007  
## 5 0.4810996  
## 6 0.5105490

# check performance using cross-validation  
gbm.perf(gbmFit,method="cv")



## [1] 43

# Predict Score and Check Model Performance  
  
library(ModelPerformance)

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,  
## logical.return = TRUE, : there is no package called 'devtools'

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,  
## logical.return = TRUE, : there is no package called 'aod'

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,  
## logical.return = TRUE, : there is no package called 'sqldf'

## This package provides Rank Ordering Table and other model performance measures.  
## Company Website: http://www.k2analytics.co.in

##   
## Attaching package: 'ModelPerformance'

## The following object is masked \_by\_ '.GlobalEnv':  
##   
## decile

train$predict.score <- predict(   
 gbmFit, newdata=train, type = "response")

## Using 43 trees...

train$decile = ModelPerformance::decile(train$predict.score)  
rank <- ModelPerformance::ROTable(train, "Attrition", "predict.score")  
  
View(rank)  
  
  
test$predict.score <- predict(   
 gbmFit, newdata=test, type = "response")

## Using 43 trees...

test$decile = ModelPerformance::decile(test$predict.score)  
rank <- ModelPerformance::ROTable(test, "Attrition", "predict.score")  
View(rank)  
  
  
?ROTable  
#Model performance on train data

ModelPerformance::concordance(train,"Attrition", "predict.score")

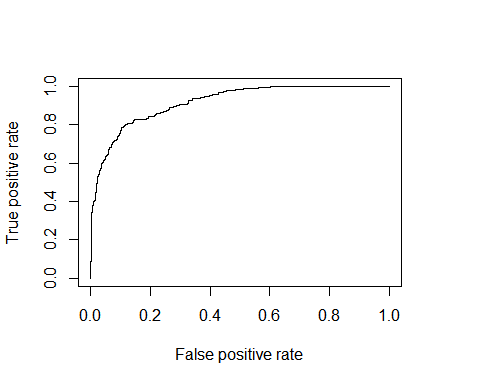
## $`Percent Concordance`  
## [1] 91.70924  
##   
## $`Percent Discordance`  
## [1] 8.290763  
##   
## $`Percent Tied`  
## [1] 0  
##   
## $Pairs  
## [1] 588245

ModelPerformance::KS\_AUC(train,"Attrition", "predict.score")

#Model performance on test data

ModelPerformance::concordance(test,"Attrition", "predict.score")

ModelPerformance::KS\_AUC(test,"Attrition", "predict.score")



## auc ks   
## 0.9170924 0.6851312

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. | Model Performance Measure | Train Data | Test Data |
| 1. | AUC | 91.68% | 87.80% |
| 2. | KS | 65.12% | 59.32% |
| 3. | Gini | 83.37% | 75.60% |

Model performance measure with GBM Model Tuning:

AUC = 91.16% Based of Train Dataset

KS =65.12%

AUC =87.80% Based of Test Dataset

KS =59.32%