Project1.R

mishrrak

2019-10-05

knitr::opts\_chunk$set(error = FALSE)  
  
Attrition <- read.csv("C:/Users/Mishrrak/Downloads/Imarticus\_Project/Attrition.csv")  
View(Attrition)  
  
# R Packagesrequired  
#library(data.table)  
#install.packages("entropy")  
#library(entropy)  
#library(ggplot2)  
#library(caTools)  
#library(ROCR)  
#library(rpart)  
#library(e1071)  
#library(rpart)  
#library(rpart.plot)  
#library(caret)  
#library(corrplot)  
#install.packages("pROC")  
#library(pROC)  
  
?options

## starting httpd help server ... done

# Checking the dimension of data  
  
dim(Attrition)

## [1] 1470 35

# Total number of Rows 1470 & Total number of column 35  
  
# Now check the null values in dataset  
  
sum(is.na(Attrition))

## [1] 0

# There is 0 null value availabe in dataset.  
  
  
#Observed the dataset & find that the attributes EmployeeNumber, Over18, EmployeeCount and StandardHours all carry the same value for each observation.therefore drop these columns.  
  
Attrition$EmployeeNumber=Attrition$Over18=Attrition$EmployeeCount=Attrition$StandardHours = NULL  
  
  
# Now we check the Types of features in dataset  
  
str(Attrition)

## 'data.frame': 1470 obs. of 31 variables:  
## $ Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...  
## $ 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 ...  
## $ 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 ...  
## $ 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 ...

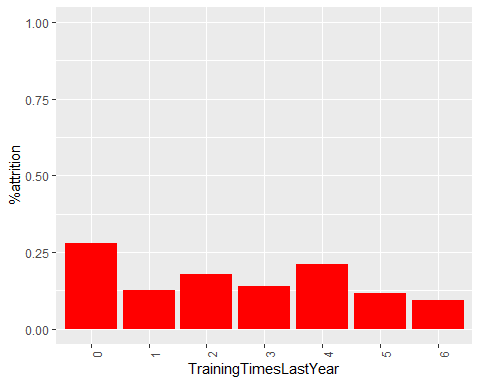
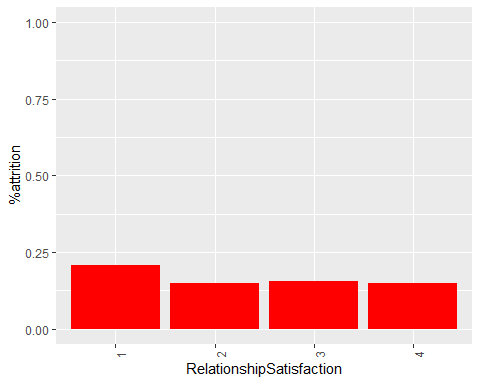
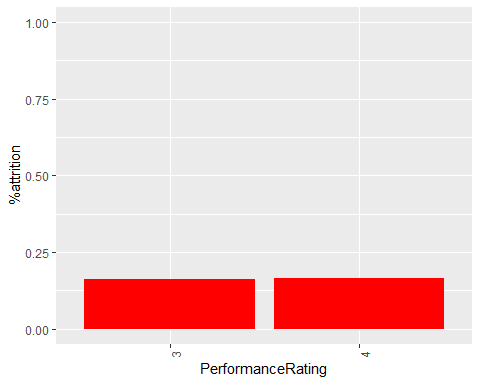
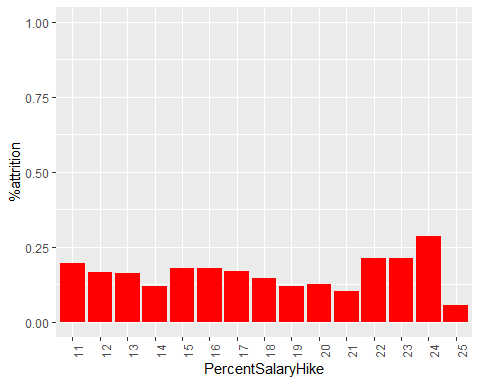
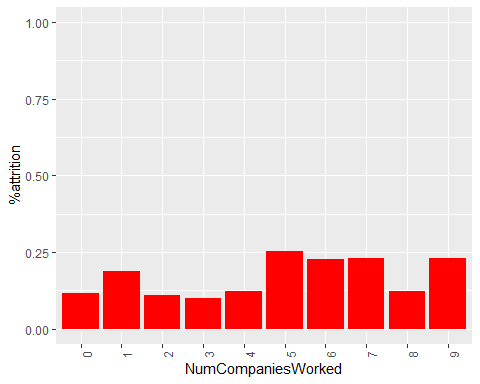
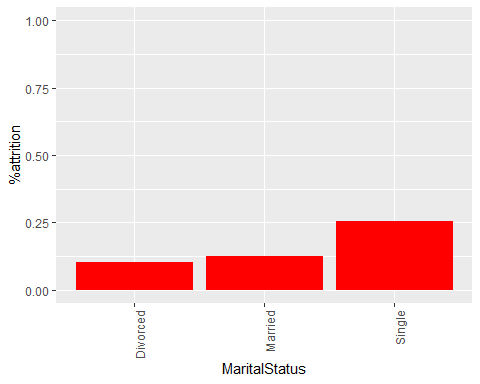
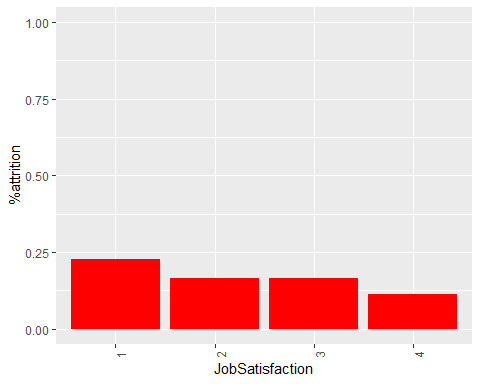
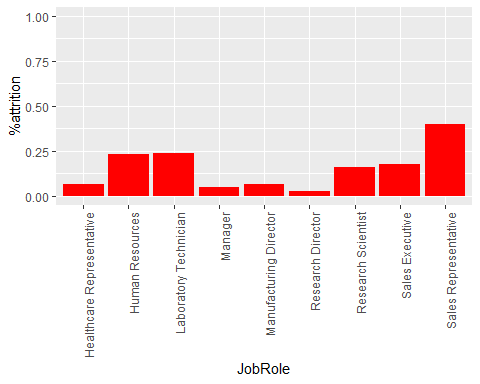
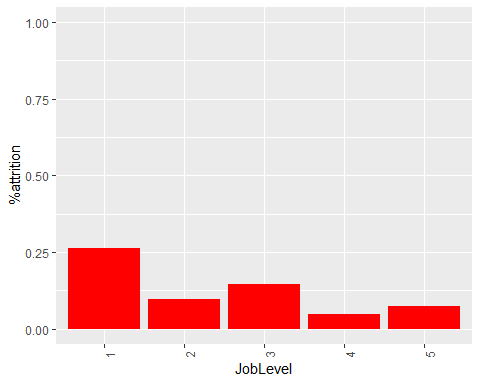
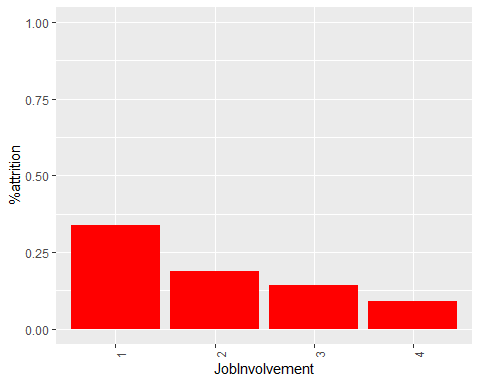
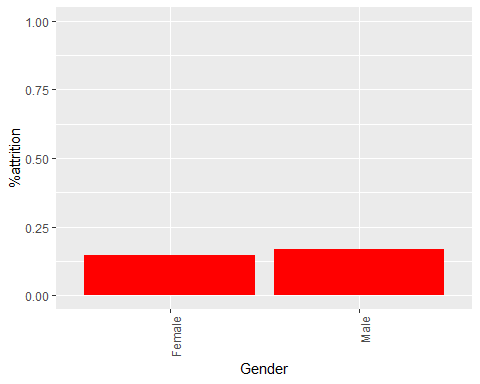
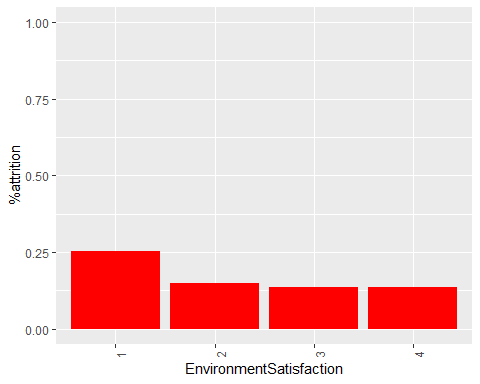
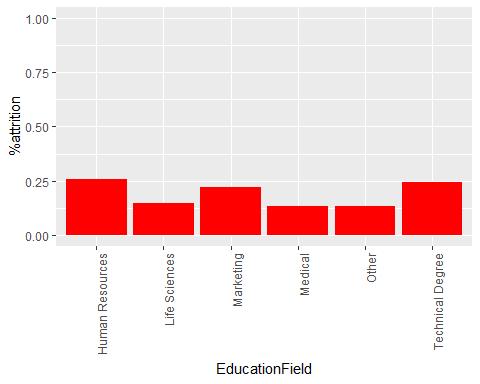
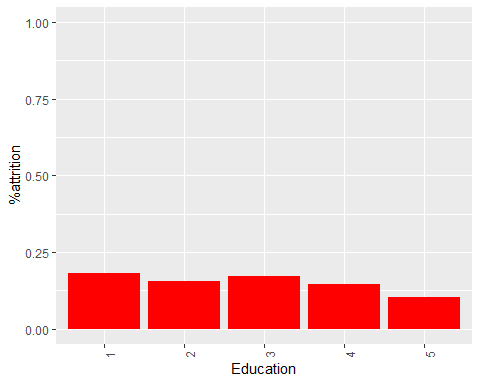
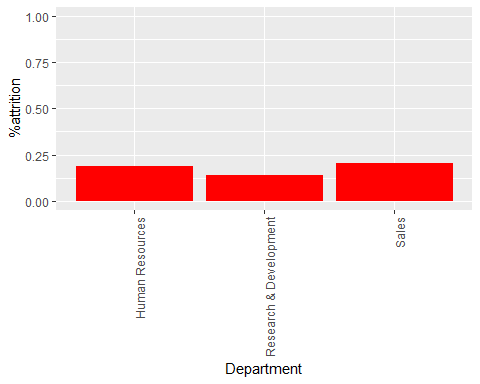
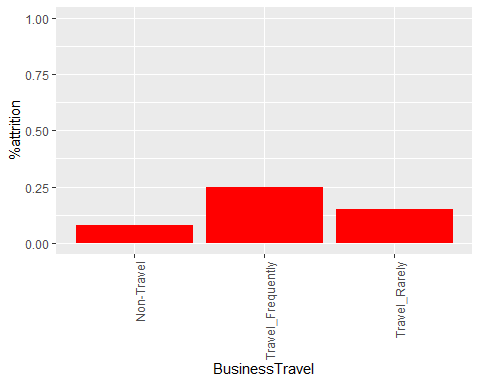
summary(Attrition)

## Age Attrition BusinessTravel DailyRate   
## Min. :18.00 No :1233 Non-Travel : 150 Min. : 102.0   
## 1st Qu.:30.00 Yes: 237 Travel\_Frequently: 277 1st Qu.: 465.0   
## Median :36.00 Travel\_Rarely :1043 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 : 63 Min. : 1.000 Min. :1.000   
## Research & Development:961 1st Qu.: 2.000 1st Qu.:2.000   
## Sales :446 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 EnvironmentSatisfaction Gender   
## Human Resources : 27 Min. :1.000 Female:588   
## Life Sciences :606 1st Qu.:2.000 Male :882   
## Marketing :159 Median :3.000   
## Medical :464 Mean :2.722   
## Other : 82 3rd Qu.:4.000   
## Technical Degree:132 Max. :4.000   
##   
## HourlyRate JobInvolvement JobLevel   
## Min. : 30.00 Min. :1.00 Min. :1.000   
## 1st Qu.: 48.00 1st Qu.:2.00 1st Qu.:1.000   
## Median : 66.00 Median :3.00 Median :2.000   
## Mean : 65.89 Mean :2.73 Mean :2.064   
## 3rd Qu.: 83.75 3rd Qu.:3.00 3rd Qu.:3.000   
## Max. :100.00 Max. :4.00 Max. :5.000   
##   
## JobRole JobSatisfaction MaritalStatus  
## Sales Executive :326 Min. :1.000 Divorced:327   
## Research Scientist :292 1st Qu.:2.000 Married :673   
## Laboratory Technician :259 Median :3.000 Single :470   
## Manufacturing Director :145 Mean :2.729   
## Healthcare Representative:131 3rd Qu.:4.000   
## Manager :102 Max. :4.000   
## (Other) :215   
## MonthlyIncome MonthlyRate NumCompaniesWorked OverTime   
## Min. : 1009 Min. : 2094 Min. :0.000 No :1054   
## 1st Qu.: 2911 1st Qu.: 8047 1st Qu.:1.000 Yes: 416   
## 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  
## Min. :11.00 Min. :3.000 Min. :1.000   
## 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   
##   
## StockOptionLevel TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  
## Min. :0.0000 Min. : 0.00 Min. :0.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.: 6.00 1st Qu.:2.000 1st Qu.:2.000   
## Median :1.0000 Median :10.00 Median :3.000 Median :3.000   
## Mean :0.7939 Mean :11.28 Mean :2.799 Mean :2.761   
## 3rd Qu.:1.0000 3rd Qu.:15.00 3rd Qu.:3.000 3rd Qu.:3.000   
## Max. :3.0000 Max. :40.00 Max. :6.000 Max. :4.000   
##   
## YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion  
## Min. : 0.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.: 3.000 1st Qu.: 2.000 1st Qu.: 0.000   
## Median : 5.000 Median : 3.000 Median : 1.000   
## Mean : 7.008 Mean : 4.229 Mean : 2.188   
## 3rd Qu.: 9.000 3rd Qu.: 7.000 3rd Qu.: 3.000   
## Max. :40.000 Max. :18.000 Max. :15.000   
##   
## YearsWithCurrManager  
## Min. : 0.000   
## 1st Qu.: 2.000   
## Median : 3.000   
## Mean : 4.123   
## 3rd Qu.: 7.000   
## Max. :17.000   
##

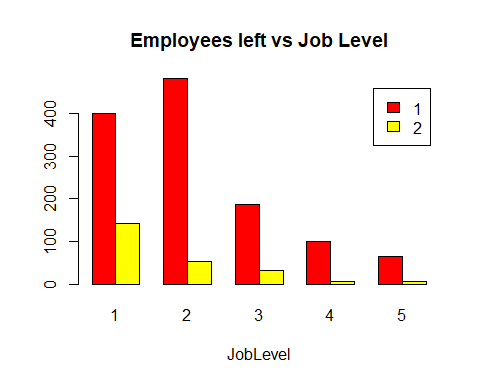
# Here we observed that there few Categorical data which needs to be factories.  
  
categorical <- c(2,3,5,7:10,12:16,19,21,22,23,26)  
#Attrition[,names] <- lapply(Attrition[,names] , factor)  
as.factor(as.numeric(Attrition$Attrition)) -> Attrition$Attrition  
  
  
# Now check summary of data to know about the distribution of data  
  
summary(Attrition)

## Age Attrition BusinessTravel DailyRate   
## Min. :18.00 1:1233 Non-Travel : 150 Min. : 102.0   
## 1st Qu.:30.00 2: 237 Travel\_Frequently: 277 1st Qu.: 465.0   
## Median :36.00 Travel\_Rarely :1043 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 : 63 Min. : 1.000 Min. :1.000   
## Research & Development:961 1st Qu.: 2.000 1st Qu.:2.000   
## Sales :446 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 EnvironmentSatisfaction Gender   
## Human Resources : 27 Min. :1.000 Female:588   
## Life Sciences :606 1st Qu.:2.000 Male :882   
## Marketing :159 Median :3.000   
## Medical :464 Mean :2.722   
## Other : 82 3rd Qu.:4.000   
## Technical Degree:132 Max. :4.000   
##   
## HourlyRate JobInvolvement JobLevel   
## Min. : 30.00 Min. :1.00 Min. :1.000   
## 1st Qu.: 48.00 1st Qu.:2.00 1st Qu.:1.000   
## Median : 66.00 Median :3.00 Median :2.000   
## Mean : 65.89 Mean :2.73 Mean :2.064   
## 3rd Qu.: 83.75 3rd Qu.:3.00 3rd Qu.:3.000   
## Max. :100.00 Max. :4.00 Max. :5.000   
##   
## JobRole JobSatisfaction MaritalStatus  
## Sales Executive :326 Min. :1.000 Divorced:327   
## Research Scientist :292 1st Qu.:2.000 Married :673   
## Laboratory Technician :259 Median :3.000 Single :470   
## Manufacturing Director :145 Mean :2.729   
## Healthcare Representative:131 3rd Qu.:4.000   
## Manager :102 Max. :4.000   
## (Other) :215   
## MonthlyIncome MonthlyRate NumCompaniesWorked OverTime   
## Min. : 1009 Min. : 2094 Min. :0.000 No :1054   
## 1st Qu.: 2911 1st Qu.: 8047 1st Qu.:1.000 Yes: 416   
## 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  
## Min. :11.00 Min. :3.000 Min. :1.000   
## 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   
##   
## StockOptionLevel TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  
## Min. :0.0000 Min. : 0.00 Min. :0.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.: 6.00 1st Qu.:2.000 1st Qu.:2.000   
## Median :1.0000 Median :10.00 Median :3.000 Median :3.000   
## Mean :0.7939 Mean :11.28 Mean :2.799 Mean :2.761   
## 3rd Qu.:1.0000 3rd Qu.:15.00 3rd Qu.:3.000 3rd Qu.:3.000   
## Max. :3.0000 Max. :40.00 Max. :6.000 Max. :4.000   
##   
## YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion  
## Min. : 0.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.: 3.000 1st Qu.: 2.000 1st Qu.: 0.000   
## Median : 5.000 Median : 3.000 Median : 1.000   
## Mean : 7.008 Mean : 4.229 Mean : 2.188   
## 3rd Qu.: 9.000 3rd Qu.: 7.000 3rd Qu.: 3.000   
## Max. :40.000 Max. :18.000 Max. :15.000   
##   
## YearsWithCurrManager  
## Min. : 0.000   
## 1st Qu.: 2.000   
## Median : 3.000   
## Mean : 4.123   
## 3rd Qu.: 7.000   
## Max. :17.000   
##

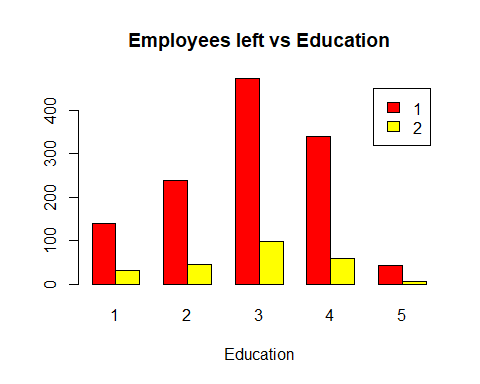
### % attrition across categorical variables  
  
  
  
freq\_tbl <- apply(Attrition[,categorical],2, function(x) table(Attrition$Attrition,x))  
freq\_tbl <- lapply(freq\_tbl,function(x) as.data.frame.matrix(x))  
  
library(ggplot2)  
  
perc\_attrition\_plot <- list()  
i =0  
for(name in names(freq\_tbl)[-1]){  
 i <- i +1  
 var\_data <- data.frame(apply(freq\_tbl[name][[1]],2, function(x) x[2]/sum(x)))  
 colnames(var\_data) <- name  
 my\_plot <- ggplot(data=var\_data, aes(x=row.names(var\_data), y=var\_data[,name])) + geom\_bar(stat="identity",fill='red') +  
 ylim(0.0,1.0) + ylab("%attrition") + xlab(name) + theme(axis.text.x = element\_text(angle = 90, hjust = 1))  
 plot(my\_plot)  
 remove(my\_plot)  
}



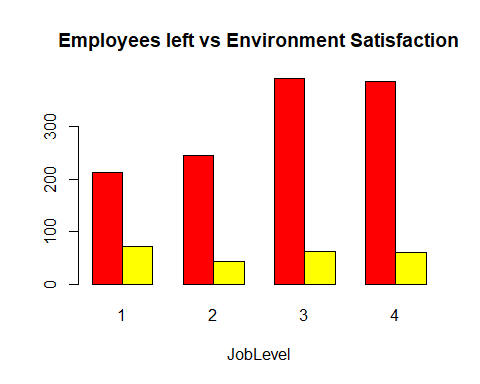
## From the above plots we see that percentage churn is more for employees who are single, do over time, travel frequently and have less worklife balance satisfaction.  
  
  
plottable1=table(Attrition$Attrition,Attrition$JobLevel)  
  
plottable2=table(Attrition$Attrition,Attrition$Education)  
  
plottable3=table(Attrition$Attrition,Attrition$EnvironmentSatisfaction)  
  
plottable4=table(Attrition$Attrition,Attrition$JobInvolvement)  
  
plottable5=table(Attrition$Attrition,Attrition$PercentSalaryHike)  
  
plottable6=table(Attrition$Attrition,Attrition$PerformanceRating)  
  
plottable7=table(Attrition$Attrition,Attrition$StockOptionLevel)  
  
plottable8=table(Attrition$Attrition,Attrition$YearsAtCompany)  
  
plottable9=table(Attrition$Attrition,Attrition$YearsInCurrentRole)  
  
plottable10 = table(Attrition$Attrition, Attrition$Department)  
  
plottable11 = table(Attrition$Attrition, Attrition$MaritalStatus)  
  
plottable12 = table(Attrition$Attrition, Attrition$Age)  
  
  
  
barplot(plottable1, main="Employees left vs Job Level", xlab="JobLevel",col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)

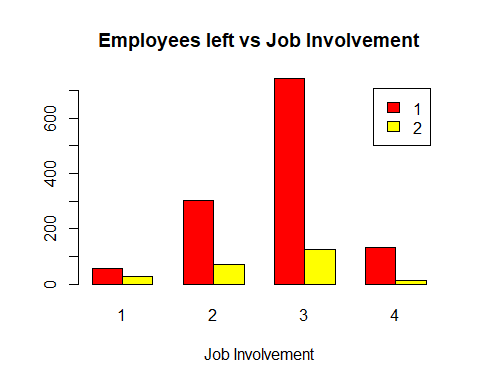


barplot(plottable2, main="Employees left vs Education", xlab="Education",col=c("Red","Yellow"),legend=rownames(plottable2),beside = TRUE)

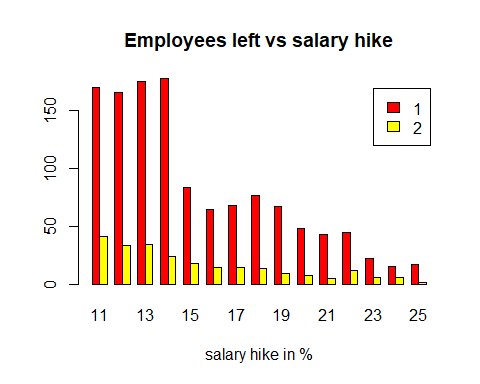


barplot(plottable3, main="Employees left vs Environment Satisfaction", xlab="JobLevel", col=c("Red","Yellow"),beside = TRUE)

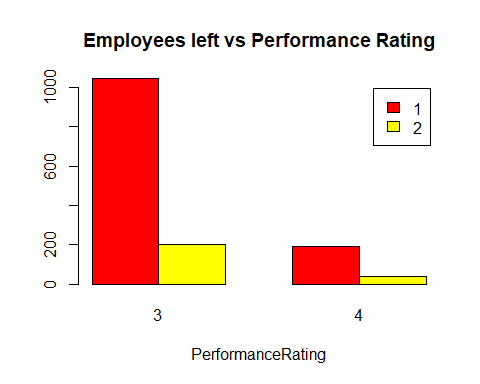
barplot(plottable4, main="Employees left vs Job Involvement", xlab="Job Involvement", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE) 



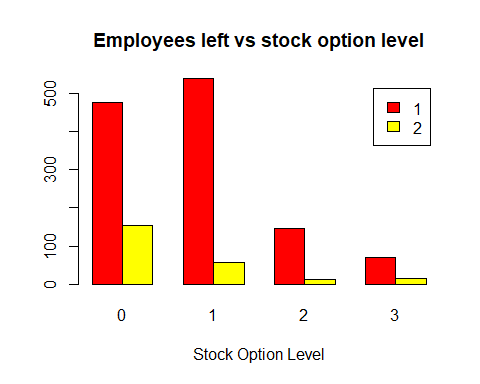
barplot(plottable5, main="Employees left vs salary hike", xlab="salary hike in %", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



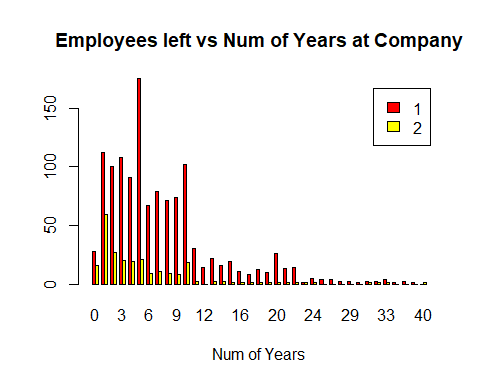
barplot(plottable6, main="Employees left vs Performance Rating", xlab="PerformanceRating",col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



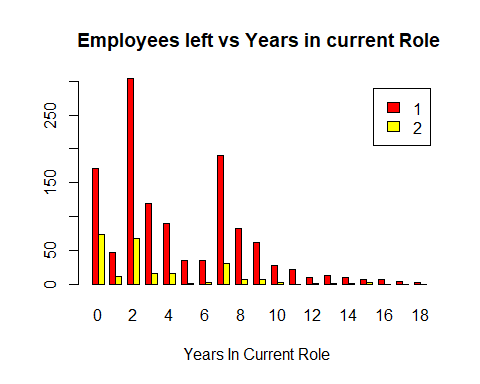
barplot(plottable7, main="Employees left vs stock option level", xlab="Stock Option Level", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



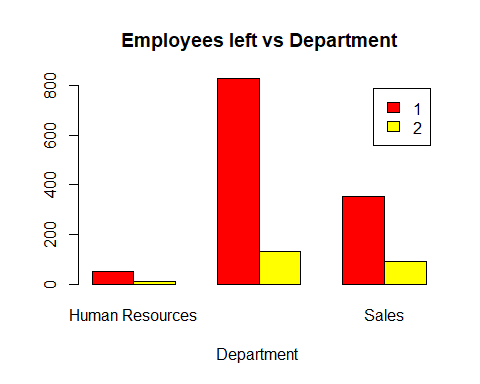
barplot(plottable8, main="Employees left vs Num of Years at Company", xlab="Num of Years", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



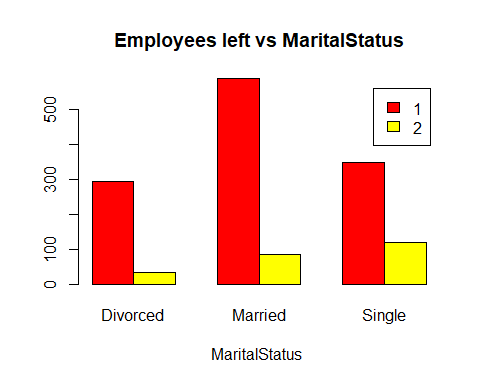
barplot(plottable9, main="Employees left vs Years in current Role", xlab="Years In Current Role ", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



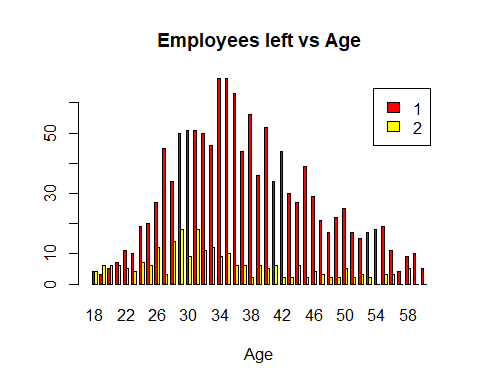
barplot(plottable10, main="Employees left vs Department", xlab="Department ", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



barplot(plottable11, main="Employees left vs MaritalStatus", xlab="MaritalStatus ", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



barplot(plottable12, main="Employees left vs Age", xlab="Age", col=c("Red","Yellow"),legend=rownames(plottable1),beside = TRUE)



#Dummy variables  
  
dummy = Attrition  
  
str(dummy)

## 'data.frame': 1470 obs. of 31 variables:  
## $ Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : Factor w/ 2 levels "1","2": 2 1 2 1 1 1 1 1 1 1 ...  
## $ 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 ...  
## $ 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 ...  
## $ 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 ...

dummy$Attrition=as.numeric(dummy$Attrition)  
  
dummy$BusinessTravel=as.numeric(dummy$BusinessTravel)  
  
dummy$Department=as.numeric(dummy$Department)  
  
dummy$EducationField=as.numeric(dummy$EducationField)  
  
dummy$Gender=as.numeric(dummy$Gender)  
  
dummy$JobRole=as.numeric(dummy$JobRole)  
  
dummy$MaritalStatus=as.numeric(dummy$MaritalStatus)  
  
dummy$OverTime=as.numeric(dummy$OverTime)  
  
dummy$DailyRate = as.numeric(dummy$DailyRate)  
  
dummy$DistanceFromHome <- as.numeric(dummy$DistanceFromHome)  
  
dummy$Education <- as.numeric(dummy$Education)  
  
dummy$EnvironmentSatisfaction <- as.numeric(dummy$EnvironmentSatisfaction)  
  
dummy$JobLevel <- as.numeric(dummy$JobLevel)  
  
dummy <- dummy[,c(2:10,13,14,16,20)]  
  
corTable=cor(dummy)  
  
corTable

## Attrition BusinessTravel DailyRate  
## Attrition 1.000000e+00 7.377695e-05 -0.056651992  
## BusinessTravel 7.377695e-05 1.000000e+00 -0.004086034  
## DailyRate -5.665199e-02 -4.086034e-03 1.000000000  
## Department 6.399060e-02 -9.043571e-03 0.007108714  
## DistanceFromHome 7.792358e-02 -2.446944e-02 -0.004985337  
## Education -3.137282e-02 7.569331e-04 -0.016806433  
## EducationField 2.684555e-02 2.372383e-02 0.037709229  
## EnvironmentSatisfaction -1.033690e-01 4.174405e-03 0.018354854  
## Gender 2.945325e-02 -3.298096e-02 -0.011716138  
## JobLevel -1.691048e-01 1.931128e-02 0.002966335  
## JobRole 6.715150e-02 2.724411e-03 -0.009472201  
## MaritalStatus 1.620702e-01 2.400051e-02 -0.069585641  
## OverTime 2.461180e-01 1.654304e-02 0.009134970  
## Department DistanceFromHome Education  
## Attrition 0.063990596 0.077923583 -0.0313728196  
## BusinessTravel -0.009043571 -0.024469442 0.0007569331  
## DailyRate 0.007108714 -0.004985337 -0.0168064332  
## Department 1.000000000 0.017224804 0.0079964220  
## DistanceFromHome 0.017224804 1.000000000 0.0210418256  
## Education 0.007996422 0.021041826 1.0000000000  
## EducationField 0.013719502 0.002013453 -0.0395921504  
## EnvironmentSatisfaction -0.019395271 -0.016075327 -0.0271283133  
## Gender -0.041583290 -0.001850528 -0.0165468274  
## JobLevel 0.101963106 0.005302731 0.1015888862  
## JobRole 0.662431198 -0.001014963 0.0042357606  
## MaritalStatus 0.056073435 -0.014437031 0.0040526543  
## OverTime 0.007480968 0.025513635 -0.0203217674  
## EducationField EnvironmentSatisfaction  
## Attrition 0.026845546 -0.1033689783  
## BusinessTravel 0.023723829 0.0041744049  
## DailyRate 0.037709229 0.0183548543  
## Department 0.013719502 -0.0193952706  
## DistanceFromHome 0.002013453 -0.0160753270  
## Education -0.039592150 -0.0271283133  
## EducationField 1.000000000 0.0431634907  
## EnvironmentSatisfaction 0.043163491 1.0000000000  
## Gender -0.002504019 0.0005083139  
## JobLevel -0.044932672 0.0012116994  
## JobRole 0.015598782 -0.0173213826  
## MaritalStatus 0.014419541 -0.0035934733  
## OverTime 0.002258600 0.0701317268  
## Gender JobLevel JobRole  
## Attrition 0.0294532532 -0.1691047509 0.067151495  
## BusinessTravel -0.0329809563 0.0193112821 0.002724411  
## DailyRate -0.0117161379 0.0029663349 -0.009472201  
## Department -0.0415832902 0.1019631058 0.662431198  
## DistanceFromHome -0.0018505280 0.0053027306 -0.001014963  
## Education -0.0165468274 0.1015888862 0.004235761  
## EducationField -0.0025040188 -0.0449326718 0.015598782  
## EnvironmentSatisfaction 0.0005083139 0.0012116994 -0.017321383  
## Gender 1.0000000000 -0.0394031027 -0.039722900  
## JobLevel -0.0394031027 1.0000000000 -0.085457434  
## JobRole -0.0397229000 -0.0854574339 1.000000000  
## MaritalStatus -0.0471825924 -0.0767694781 0.067956618  
## OverTime -0.0419243480 0.0005440478 0.040662366  
## MaritalStatus OverTime  
## Attrition 0.162070235 0.2461179942  
## BusinessTravel 0.024000511 0.0165430420  
## DailyRate -0.069585641 0.0091349699  
## Department 0.056073435 0.0074809676  
## DistanceFromHome -0.014437031 0.0255136349  
## Education 0.004052654 -0.0203217674  
## EducationField 0.014419541 0.0022585999  
## EnvironmentSatisfaction -0.003593473 0.0701317268  
## Gender -0.047182592 -0.0419243480  
## JobLevel -0.076769478 0.0005440478  
## JobRole 0.067956618 0.0406623662  
## MaritalStatus 1.000000000 -0.0175213816  
## OverTime -0.017521382 1.0000000000

#A correlation plot  
  
#corrplot( cor(as.matrix(dummy), method = "pearson", use = "complete.obs") ,is.corr = FALSE, type = "lower", order = "hclust", tl.col = "black", tl.srt = 360)  
  
#After carefully examining the correlation plot and table, find that there are a lot of correlated features.  
  
# We run the Boruta to examine which features are to be dropped from the pair of each correlated features.  
  
library(Boruta)

## Loading required package: ranger

A1 <- Boruta(Attrition~., data = Attrition)  
  
A1$finalDecision

## Age BusinessTravel DailyRate   
## Confirmed Rejected Rejected   
## Department DistanceFromHome Education   
## Tentative Tentative Rejected   
## EducationField EnvironmentSatisfaction Gender   
## Rejected Confirmed Rejected   
## HourlyRate JobInvolvement JobLevel   
## Rejected Confirmed Confirmed   
## JobRole JobSatisfaction MaritalStatus   
## Confirmed Confirmed Confirmed   
## MonthlyIncome MonthlyRate NumCompaniesWorked   
## Confirmed Rejected Confirmed   
## OverTime PercentSalaryHike PerformanceRating   
## Confirmed Rejected Rejected   
## RelationshipSatisfaction StockOptionLevel TotalWorkingYears   
## Rejected Confirmed Confirmed   
## TrainingTimesLastYear WorkLifeBalance YearsAtCompany   
## Rejected Confirmed Confirmed   
## YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager   
## Confirmed Tentative Confirmed   
## Levels: Tentative Confirmed Rejected

# After using Boruta we came to know that there are total 16 significant features(Age,JobRole,JobSatisfaction,NumCompaniesWorked etc), so we'll take these features only for future.  
  
# Now we drop all other features & keep only those who are significant for Attrition  
  
Atr <- Attrition[, c(1,2,9,12:17,19,20,24,25,27,28,29,31)]  
summary(Atr)

## Age Attrition EnvironmentSatisfaction JobInvolvement  
## Min. :18.00 1:1233 Min. :1.000 Min. :1.00   
## 1st Qu.:30.00 2: 237 1st Qu.:2.000 1st Qu.:2.00   
## Median :36.00 Median :3.000 Median :3.00   
## Mean :36.92 Mean :2.722 Mean :2.73   
## 3rd Qu.:43.00 3rd Qu.:4.000 3rd Qu.:3.00   
## Max. :60.00 Max. :4.000 Max. :4.00   
##   
## JobLevel JobRole JobSatisfaction  
## Min. :1.000 Sales Executive :326 Min. :1.000   
## 1st Qu.:1.000 Research Scientist :292 1st Qu.:2.000   
## Median :2.000 Laboratory Technician :259 Median :3.000   
## Mean :2.064 Manufacturing Director :145 Mean :2.729   
## 3rd Qu.:3.000 Healthcare Representative:131 3rd Qu.:4.000   
## Max. :5.000 Manager :102 Max. :4.000   
## (Other) :215   
## MaritalStatus MonthlyIncome NumCompaniesWorked OverTime   
## Divorced:327 Min. : 1009 Min. :0.000 No :1054   
## Married :673 1st Qu.: 2911 1st Qu.:1.000 Yes: 416   
## Single :470 Median : 4919 Median :2.000   
## Mean : 6503 Mean :2.693   
## 3rd Qu.: 8379 3rd Qu.:4.000   
## Max. :19999 Max. :9.000   
##   
## StockOptionLevel TotalWorkingYears WorkLifeBalance YearsAtCompany   
## Min. :0.0000 Min. : 0.00 Min. :1.000 Min. : 0.000   
## 1st Qu.:0.0000 1st Qu.: 6.00 1st Qu.:2.000 1st Qu.: 3.000   
## Median :1.0000 Median :10.00 Median :3.000 Median : 5.000   
## Mean :0.7939 Mean :11.28 Mean :2.761 Mean : 7.008   
## 3rd Qu.:1.0000 3rd Qu.:15.00 3rd Qu.:3.000 3rd Qu.: 9.000   
## Max. :3.0000 Max. :40.00 Max. :4.000 Max. :40.000   
##   
## YearsInCurrentRole YearsWithCurrManager  
## Min. : 0.000 Min. : 0.000   
## 1st Qu.: 2.000 1st Qu.: 2.000   
## Median : 3.000 Median : 3.000   
## Mean : 4.229 Mean : 4.123   
## 3rd Qu.: 7.000 3rd Qu.: 7.000   
## Max. :18.000 Max. :17.000   
##

# Now we spliting data into training and testing using Stratified sampling  
  
set.seed(3031)  
Sample1 <- sample(2,nrow(Atr),replace = TRUE, prob = c(.7,.3))  
  
Atr\_Train <- Atr[Sample1==1,]  
Atr\_Test <- Atr[Sample1==2,]  
  
table(Atr\_Train$Attrition)

##   
## 1 2   
## 897 162

#Implementing Machine Learning Algorithms  
  
Atr\_Log =glm(Attrition~., Atr\_Train,family = binomial)  
summary(Atr\_Log)

##   
## Call:  
## glm(formula = Attrition ~ ., family = binomial, data = Atr\_Train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9619 -0.5363 -0.3152 -0.1521 3.2046   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.454e+00 1.050e+00 1.385 0.166198   
## Age -4.546e-02 1.614e-02 -2.817 0.004851 \*\*   
## EnvironmentSatisfaction -2.943e-01 9.103e-02 -3.233 0.001223 \*\*   
## JobInvolvement -5.034e-01 1.375e-01 -3.660 0.000252 \*\*\*  
## JobLevel -1.036e-01 3.418e-01 -0.303 0.761745   
## JobRoleHuman Resources 1.321e+00 6.221e-01 2.123 0.033766 \*   
## JobRoleLaboratory Technician 1.487e+00 5.297e-01 2.807 0.004995 \*\*   
## JobRoleManager -1.260e+00 9.377e-01 -1.344 0.178867   
## JobRoleManufacturing Director 1.832e-01 5.757e-01 0.318 0.750304   
## JobRoleResearch Director -1.400e+00 9.831e-01 -1.425 0.154295   
## JobRoleResearch Scientist 6.533e-01 5.408e-01 1.208 0.227009   
## JobRoleSales Executive 9.608e-01 4.611e-01 2.084 0.037198 \*   
## JobRoleSales Representative 2.033e+00 6.002e-01 3.388 0.000704 \*\*\*  
## JobSatisfaction -3.838e-01 9.129e-02 -4.204 2.62e-05 \*\*\*  
## MaritalStatusMarried 6.327e-01 3.379e-01 1.873 0.061130 .   
## MaritalStatusSingle 1.216e+00 4.114e-01 2.956 0.003117 \*\*   
## MonthlyIncome 9.270e-05 8.769e-05 1.057 0.290424   
## NumCompaniesWorked 1.357e-01 4.422e-02 3.069 0.002146 \*\*   
## OverTimeYes 1.653e+00 2.122e-01 7.793 6.56e-15 \*\*\*  
## StockOptionLevel -3.040e-01 1.772e-01 -1.716 0.086216 .   
## TotalWorkingYears -3.709e-02 3.310e-02 -1.120 0.262615   
## WorkLifeBalance -2.854e-01 1.391e-01 -2.052 0.040174 \*   
## YearsAtCompany 1.263e-01 3.995e-02 3.163 0.001563 \*\*   
## YearsInCurrentRole -9.044e-02 4.681e-02 -1.932 0.053372 .   
## YearsWithCurrManager -1.265e-01 5.124e-02 -2.469 0.013556 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 906.15 on 1058 degrees of freedom  
## Residual deviance: 679.84 on 1034 degrees of freedom  
## AIC: 729.84  
##   
## Number of Fisher Scoring iterations: 6

# Predicting the Test data  
  
Atr\_pre <- predict(Atr\_Log, Atr\_Test, type="response")  
  
  
Atr\_act <- data.frame(Atr\_pre,Atr\_Test$Attrition)  
  
colnames(Atr\_act) <- c('pred', 'actual')  
  
# THRESHOLDING : Outcome of a logistic regression model is a probability. Often, we want to make an actual prediction. We can convert the probabilities to predictions using what's called a threshold value, t. If the probability of attrition is greater than this threshold value, t, we predict that employee will churn. But if the probability of attrition is less than the threshold value, t, then we predict that employee will not churn.  
  
# If we pick a large threshold value t,then we will predict actual churn rarely, since the probability of attrition has to be really large to be greater than the threshold. This means that we will make more errors where we say that employee will not churn , but it's actually churning case.  
  
# On the other hand, if the threshold value, t, is small,we predict churn (Yes) frequently, and we predict non churn (No) rarely. This means that we will make more errors where we say that employee will not churn, but it's actually that employee will churn.  
  
#A model with a higher threshold will have a lower sensitivity and a higher specificity. A model with a lower threshold will have a higher sensitivity and a lower specificity.  
  
  
# Threshold - 0.1  
  
print("Confusion matrix for threshold 0.1")

## [1] "Confusion matrix for threshold 0.1"

thershold= 0.1  
  
confusion\_mat <- table(Atr\_Test$Attrition, Atr\_pre > thershold)  
confusion\_mat

##   
## FALSE TRUE  
## 1 204 132  
## 2 13 62

# FALSE TRUE  
# 1 219 117  
# 2 14 61  
  
# Accuracy  
  
acc <- (sum(diag(confusion\_mat))/sum(confusion\_mat)) \* 100  
acc # 68.126

## [1] 64.72019

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp <- confusion\_mat[4]  
tp\_plus\_fn <- confusion\_mat[4] + confusion\_mat[2]  
  
sensitivity <- tp/tp\_plus\_fn  
print(c("sensitivity",sensitivity)) # 0.8133

## [1] "sensitivity" "0.826666666666667"

# specificity tnr--> specificity = tn/(tn+FP)  
tn <- confusion\_mat[1]  
tn\_plus\_fp <- confusion\_mat[1] + confusion\_mat[3]  
  
specificity <- tn/tn\_plus\_fp  
print(c("specificity",specificity)) # 0.6517

## [1] "specificity" "0.607142857142857"

# Threshold - 0.3  
  
print("Confusion matrix for threshold 0.3")

## [1] "Confusion matrix for threshold 0.3"

thershold= 0.3  
  
confusion\_mat1 <- table(Atr\_Test$Attrition, Atr\_pre > thershold)  
confusion\_mat1

##   
## FALSE TRUE  
## 1 303 33  
## 2 33 42

# FALSE TRUE  
# 1 304 32  
# 2 32 43  
  
acc1 <- (sum(diag(confusion\_mat1))/sum(confusion\_mat1)) \* 100  
acc1 # 84.428

## [1] 83.94161

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp1 <- confusion\_mat1[4]  
tp\_plus\_fn1 <- confusion\_mat1[4] + confusion\_mat1[2]  
  
sensitivity1 <- tp1/tp\_plus\_fn1  
sensitivity1 # 0.5733

## [1] 0.56

# specificity tnr--> specificity = tn/(tn+FP)  
tn1 <- confusion\_mat1[1]  
tn\_plus\_fp1 <- confusion\_mat1[1] + confusion\_mat1[3]  
  
specificity1 <- tn1/tn\_plus\_fp1  
specificity1 # 0.9047

## [1] 0.9017857

# Threshold - 0.5  
  
print("Confusion matrix for threshold 0.5")

## [1] "Confusion matrix for threshold 0.5"

thershold= 0.5  
  
confusion\_mat2 <- table(Atr\_Test$Attrition, Atr\_pre > thershold)  
confusion\_mat2

##   
## FALSE TRUE  
## 1 329 7  
## 2 44 31

# FALSE TRUE  
# 1 322 14  
# 2 43 32  
  
acc2 <- (sum(diag(confusion\_mat2))/sum(confusion\_mat2)) \* 100  
acc2 # 86.131

## [1] 87.59124

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp2 <- confusion\_mat2[4]  
tp\_plus\_fn2 <- confusion\_mat2[4] + confusion\_mat2[2]  
  
sensitivity2 <- tp2/tp\_plus\_fn2  
sensitivity2 # 0.4266

## [1] 0.4133333

# specificity tnr--> specificity = tn/(tn+FP)  
tn2 <- confusion\_mat2[1]  
tn\_plus\_fp2 <- confusion\_mat2[1] + confusion\_mat2[3]  
  
specificity2 <- tn2/tn\_plus\_fp2  
specificity2 # 0.9583

## [1] 0.9791667

## Plotting Receiver operator characteristics curve to decide better on threshold  
#rocr\_pred\_logistic\_best\_treshold = prediction(Atr\_pre ,Atr\_Test$Attrition)  
#ocr\_perf\_logistic\_best\_treshold = performance(rocr\_pred\_logistic\_best\_treshold,'tpr','fpr')  
#plot(rocr\_perf\_logistic\_best\_treshold,colorize=TRUE,print.cutoffs.at = seq(0,1,.1),text.adj =c(-0.2,1.7))  
  
  
## From this plot we can say that 0.3 is the best threshold. Now we will evaluate model performance witn this threshold.  
  
  
thershold\_best\_log = 0.3  
  
conf\_mat\_log <- table(Atr\_Test$Attrition ,Atr\_pre > thershold\_best\_log)  
conf\_mat\_log

##   
## FALSE TRUE  
## 1 303 33  
## 2 33 42

#Accuracy  
ac <- (sum(diag(conf\_mat\_log))/sum(conf\_mat\_log)) \* 100  
ac

## [1] 83.94161

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tpa <- conf\_mat\_log[4]  
tp\_plus\_fna <- conf\_mat\_log[4] + conf\_mat\_log[2]  
  
sensitivitya <- tpa/tp\_plus\_fna  
sensitivitya

## [1] 0.56

# specificity tnr--> specificity = tn/(tn+FP)  
tna <- conf\_mat\_log[1]  
tn\_plus\_fpa <- conf\_mat\_log[1] + conf\_mat\_log[3]  
  
specificitya <- tna/tn\_plus\_fpa  
specificitya

## [1] 0.9017857

# 'Confusion matrix for threshold   
  
# FALSE TRUE  
# 1 304 32  
# 2 32 43

Model Performance’

#[1] "Accuracy" "84.428"  
#[1] "sensitivity" "0.5733"  
#[1] "specificity" "0.9047"  
  
  
## Now we try to reduce the complexity of the model by selecting important features for the model based on p-value. Lower the value impotant the feature is.  
  
Atr\_Log1=glm(Attrition~ Age+EnvironmentSatisfaction+JobInvolvement+JobSatisfaction+OverTime+WorkLifeBalance+NumCompaniesWorked+StockOptionLevel+YearsWithCurrManager+JobRole+JobLevel , Atr\_Train,family = binomial)  
  
Atr\_pre1 <- predict(Atr\_Log1, Atr\_Test, type="response")  
  
Atr\_act1 <- data.frame(Atr\_pre1,Atr\_Test$Attrition)  
  
colnames(Atr\_act1) <- c('pred', 'actual')  
  
library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

actual1 <- mutate(Atr\_act1,pred = ifelse(pred<=.3,1,2))  
  
taba <- table(actual1$pred, actual1$actual)  
taba

##   
## 1 2  
## 1 302 31  
## 2 34 44

acc4 <- (sum(diag(taba))/sum(taba)) \* 100  
acc4 # 84.42822

## [1] 84.18491

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp4 <- taba[4]  
tp\_plus\_fn4 <- taba[4] + taba[2]  
  
sensitivity4 <- tp4/tp\_plus\_fn4  
sensitivity4 #0.5679012

## [1] 0.5641026

# specificity tnr--> specificity = tn/(tn+FP)  
tn4 <- taba[1]  
tn\_plus\_fp4 <- taba[1] + taba[3]  
  
specificity4 <- tn4/tn\_plus\_fp4  
specificity4 # 0.9121212

## [1] 0.9069069

# 'Confusion matrix for threshold   
  
# 1 2  
# 1 301 29  
# 2 35 46  
  
'Model Performance'

## [1] "Model Performance"

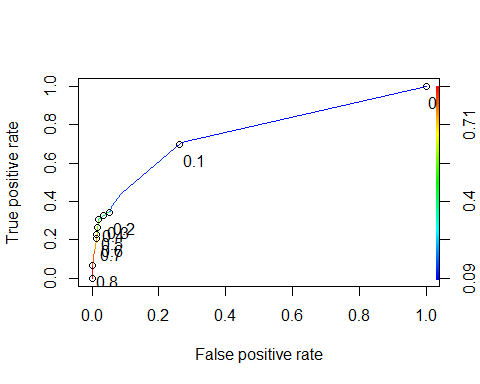
#[1] "Accuracy" "84.428"  
#[1] "sensitivity" "0.5679012"  
#[1] "specificity" "0.9121212"  
  
  
  
  
  
##### Decision Tree  
library(rpart)  
Atr\_Dt <- rpart(Attrition ~ ., method="class", data= Atr\_Train)  
  
Dt\_pre = as.data.frame.matrix(predict(Atr\_Dt,newdata = Atr\_Test,type = "prob"))  
Dt\_pre <- Dt\_pre$`2`  
  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

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

rocr\_Dt\_pre = prediction(Dt\_pre ,Atr\_Test$Attrition)  
rocr\_perf\_Dt = performance(rocr\_Dt\_pre,'tpr','fpr')  
plot(rocr\_perf\_Dt,colorize=TRUE,print.cutoffs.at = seq(0,1,.1),text.adj =c(-0.2,1.7))



# We see that threshold 0.3 is the best threshold here.  
  
Atr\_act2 <- data.frame(Dt\_pre,Atr\_Test$Attrition)  
  
colnames(Atr\_act2) <- c('pred', 'actual')  
  
library(dplyr)  
actual2 <- mutate(Atr\_act2,pred = ifelse(pred<=.3,1,2))  
  
taba1 <- table(actual2$pred, actual2$actual)  
taba1

##   
## 1 2  
## 1 330 52  
## 2 6 23

acc5 <- (sum(diag(taba1))/sum(taba1)) \*100  
acc5 #85.15815

## [1] 85.88808

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp5 <- taba1[4]  
tp\_plus\_fn5 <- taba1[4] + taba1[2]  
  
sensitivity5 <- tp5/tp\_plus\_fn5  
sensitivity5 #0.7058824

## [1] 0.7931034

# specificity tnr--> specificity = tn/(tn+FP)  
tn5 <- taba1[1]  
tn\_plus\_fp5 <- taba1[1] + taba1[3]  
  
specificity5 <- tn5/tn\_plus\_fp5  
specificity5 # 0.8647215

## [1] 0.8638743

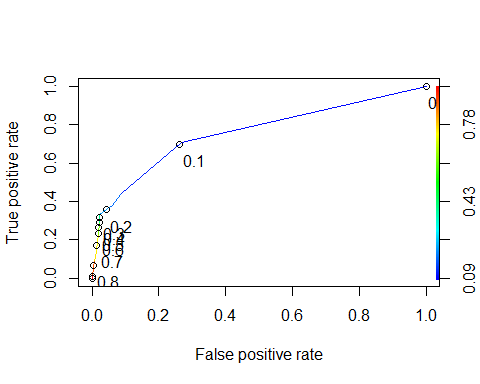
# 'Confusion matrix for threshold   
  
# 1 2  
# 1 326 51  
# 2 10 24  
  
'Model Performance'

## [1] "Model Performance"

#[1] "Accuracy" "85.15815"  
#[1] "sensitivity" "0.7058824"  
#[1] "specificity" "0.8647215"  
  
  
  
# Top 10 important variables  
sort(Atr\_Dt$variable.importance,decreasing = TRUE)[1:10]

## TotalWorkingYears MonthlyIncome OverTime   
## 18.918788 18.193385 14.037172   
## MaritalStatus Age JobRole   
## 11.118040 11.077660 10.513686   
## StockOptionLevel YearsInCurrentRole EnvironmentSatisfaction   
## 7.417346 3.820114 3.099144   
## YearsAtCompany   
## 3.000370

# We can observe that 6 out of 10 imp variables are more significance, so we'll process with those 6 variables.  
  
  
Dt\_imp\_vars <- names(sort(Atr\_Dt$variable.importance,decreasing = TRUE)[1:6])  
  
  
  
Art\_Dt1 <- rpart(Attrition ~ ., method="class", data=Atr\_Train[,c("Attrition",Dt\_imp\_vars)])  
  
  
Dt\_pre1 = as.data.frame.matrix(predict(Art\_Dt1,newdata = Atr\_Test[,c("Attrition",Dt\_imp\_vars)],type = "prob"))  
  
Dt\_pre1 <- Dt\_pre1$`2`  
  
rocr\_Dt\_pre1 = prediction(Dt\_pre1 ,Atr\_Test$Attrition)  
rocr\_perf\_Dt1 = performance(rocr\_Dt\_pre1,'tpr','fpr')  
plot(rocr\_perf\_Dt1,colorize=TRUE,print.cutoffs.at = seq(0,1,.1),text.adj =c(-0.2,1.7))



# We see that threshold 0.2 is the best threshold here.  
  
threshold\_Dt\_pre1 <- 0.2  
  
taba2 <- table(Atr\_Test$Attrition ,Dt\_pre1 > threshold\_Dt\_pre1)  
taba2

##   
## FALSE TRUE  
## 1 329 7  
## 2 50 25

#Accuracy  
  
acc6 <- (sum(diag(taba2))/sum(taba2)) \*100  
acc6 #86.13139

## [1] 86.13139

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp6 <- taba2[4]  
tp\_plus\_fn6 <- taba2[4] + taba2[2]  
  
sensitivity6 <- tp6/tp\_plus\_fn6  
sensitivity6 # 0.3333333

## [1] 0.3333333

# specificity tnr--> specificity = tn/(tn+FP)  
tn6 <- taba2[1]  
tn\_plus\_fp6 <- taba2[1] + taba2[3]  
  
specificity6 <- tn6/tn\_plus\_fp6  
specificity6 # 0.9791667

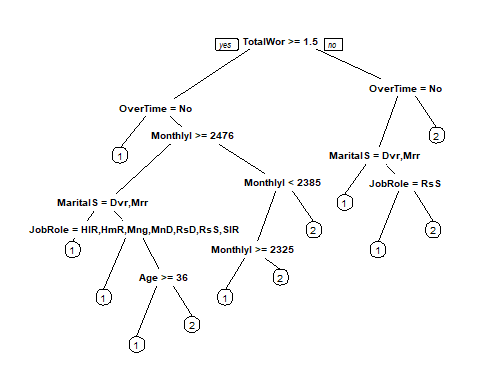
## [1] 0.9791667

# 'Confusion matrix for threshold   
  
# 1 2  
# 1 329 7  
# 2 50 25  
  
'Model Performance'

## [1] "Model Performance"

#[1] "Accuracy" "86.13139"  
#[1] "sensitivity" "0.3333333"  
#[1] "specificity" "0.9791667"  
  
library(rpart.plot)  
prp(Art\_Dt1)

## Warning: Bad 'data' field in model 'call' (expected a data.frame or a matrix).  
## To silence this warning:  
## Call prp with roundint=FALSE,  
## or rebuild the rpart model with model=TRUE.



####################################  
  
## Random Forest  
  
library(randomForest)

## randomForest 4.6-14

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

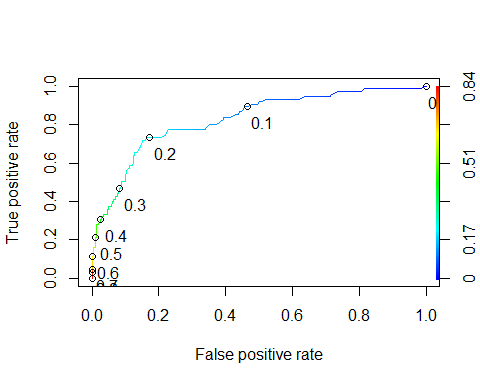
##   
## Attaching package: 'randomForest'

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

## The following object is masked from 'package:ranger':  
##   
## importance

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

Atr\_RF <- randomForest(Attrition ~ ., Atr\_Train, ntree = 500)  
  
RF\_pre <- as.data.frame.matrix(predict(Atr\_RF,Atr\_Test,type = "prob"))  
  
RF\_pre <- RF\_pre$`2`  
  
rocr\_RF\_pre = prediction(RF\_pre ,Atr\_Test$Attrition)  
rocr\_perf\_RF1 = performance(rocr\_RF\_pre,'tpr','fpr')  
plot(rocr\_perf\_RF1,colorize=TRUE,print.cutoffs.at = seq(0,1,.1),text.adj =c(-0.2,1.7))



# We see that threshold 0.3 is the best threshold here.  
  
threshold\_RF\_pre1 <- 0.3  
  
taba3 <- table(Atr\_Test$Attrition ,RF\_pre > threshold\_RF\_pre1)  
taba3

##   
## FALSE TRUE  
## 1 308 28  
## 2 41 34

#Accuracy  
  
acc7 <- (sum(diag(taba3))/sum(taba3)) \*100  
acc7 #84.18491

## [1] 83.21168

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp7 <- taba3[4]  
tp\_plus\_fn7 <- taba3[4] + taba3[2]  
  
sensitivity7 <- tp7/tp\_plus\_fn7  
sensitivity7 # 0.453333

## [1] 0.4533333

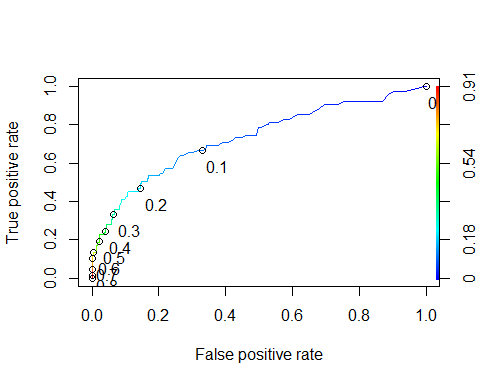
# specificity tnr--> specificity = tn/(tn+FP)  
tn7 <- taba3[1]  
tn\_plus\_fp7 <- taba3[1] + taba3[3]  
  
specificity7 <- tn7/tn\_plus\_fp7  
specificity7 # 0.9285714

## [1] 0.9166667

# 'Confusion matrix for threshold   
  
# 1 2  
# 1 312 24  
# 2 41 34  
  
'Model Performance'

## [1] "Model Performance"

#[1] "Accuracy" "84.18491"  
#[1] "sensitivity" "0.453333"  
#[1] "specificity" "0.9285714"  
  
  
# RF With imp variables  
  
Atr\_RF1 <- randomForest(Attrition ~ ., method="class", data=Atr\_Train[,c("Attrition",Dt\_imp\_vars)], ntree = 500)  
  
RF\_pre1 <- as.data.frame.matrix(predict(Atr\_RF1,newdata = Atr\_Test[,c("Attrition",Dt\_imp\_vars)],type = "prob"))  
  
RF\_pre1 <- RF\_pre1$`2`  
  
rocr\_RF\_pre1 = prediction(RF\_pre1 ,Atr\_Test$Attrition)  
rocr\_perf\_RF2 = performance(rocr\_RF\_pre1,'tpr','fpr')  
plot(rocr\_perf\_RF2,colorize=TRUE,print.cutoffs.at = seq(0,1,.1),text.adj =c(-0.2,1.7))



# We see that threshold 0.3 is the best threshold here.  
  
threshold\_RF\_pre1 <- 0.3  
  
taba5 <- table(Atr\_Test$Attrition ,RF\_pre1 > threshold\_RF\_pre1)  
taba5

##   
## FALSE TRUE  
## 1 315 21  
## 2 50 25

#Accuracy  
  
acc9 <- (sum(diag(taba5))/sum(taba5)) \*100  
acc9 #83.21168

## [1] 82.72506

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp9 <- taba5[4]  
tp\_plus\_fn9 <- taba5[4] + taba5[2]  
  
sensitivity9 <- tp9/tp\_plus\_fn9  
sensitivity9 # 0.36

## [1] 0.3333333

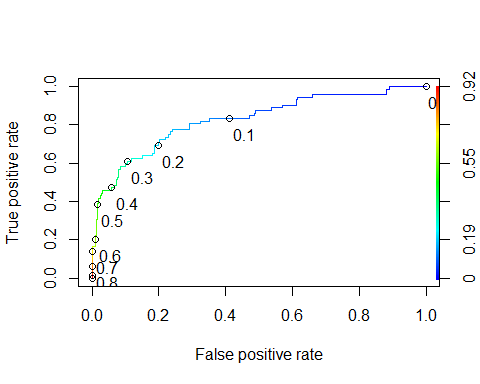
# specificity tnr--> specificity = tn/(tn+FP)  
tn9 <- taba5[1]  
tn\_plus\_fp9 <- taba5[1] + taba5[3]  
  
specificity9 <- tn9/tn\_plus\_fp9  
specificity9 # 0.9375

## [1] 0.9375

# 'Confusion matrix for threshold   
  
# 1 2  
# 1 311 25  
# 2 45 30  
  
'Model Performance'

## [1] "Model Performance"

#[1] "Accuracy" "83.21168"  
#[1] "sensitivity" "0.36"  
#[1] "specificity" "0.9375"  
  
  
#############################################  
  
## Now let us try and improve the model by engineering new features.  
Attrition1 <- read.csv("C:/Users/Mishrrak/Downloads/Imarticus\_Project/Attrition.csv")  
  
#Atr1 <- mutate(Atr,is\_first\_company=ifelse(NumCompaniesWorked == 1,1,2))  
#dim(Atr1)  
Atr1<- mutate(Atr, loyalty=(Atr$YearsAtCompany)/(Atr$TotalWorkingYears))  
  
## lower the number more volatile is the employee  
  
volatility = (Attrition1$TotalWorkingYears)/(Attrition1$NumCompaniesWorked)  
volatility[which(is.infinite(volatility))] <- Attrition1$TotalWorkingYears[which(is.infinite(volatility))]  
  
Atr1$volatility <- volatility  
  
Atr1 <- as.data.frame(Atr1)  
  
#Atr1$is\_first\_company <- factor(Atr1$is\_first\_company)  
  
set.seed(3031)  
Sample1 <- sample(2,nrow(Atr1),replace = TRUE, prob = c(.7,.3))  
  
New\_Atr\_Train <- Atr1[Sample1==1,]  
New\_Atr\_Test <- Atr1[Sample1==2,]  
  
# Modeling with new features  
  
Atr\_Log2=glm(Attrition~ Age+EnvironmentSatisfaction+JobInvolvement+JobSatisfaction+OverTime+WorkLifeBalance+NumCompaniesWorked+StockOptionLevel+YearsWithCurrManager+JobRole+JobLevel+loyalty+volatility , New\_Atr\_Train,family = binomial)  
  
  
Atr\_pre4 <- predict(Atr\_Log2, New\_Atr\_Test, type = 'response')  
  
Atr\_act4 <- data.frame(Atr\_pre4,New\_Atr\_Test$Attrition)  
  
colnames(Atr\_act4) <- c('pred', 'actual')  
  
rocr\_Atr\_pre4 = prediction(Atr\_pre4 ,New\_Atr\_Test$Attrition)  
rocr\_perf\_pre4 = performance(rocr\_Atr\_pre4,'tpr','fpr')  
plot(rocr\_perf\_pre4,colorize=TRUE,print.cutoffs.at = seq(0,1,.1),text.adj =c(-0.2,1.7))



threshold\_pre4 = 0.3  
  
taba4 <- table(New\_Atr\_Test$Attrition ,Atr\_pre4 > threshold\_pre4)  
taba4

##   
## FALSE TRUE  
## 1 300 35  
## 2 28 44

#Accuracy  
  
acc8 <- (sum(diag(taba4))/sum(taba4)) \*100  
acc8 #81.26521

## [1] 84.52088

# sensitivity tpr --> sensitivity = tp/(tp+FN)  
tp8 <- taba4[4]  
tp\_plus\_fn8 <- taba4[4] + taba4[2]  
  
sensitivity8 <- tp8/tp\_plus\_fn8  
sensitivity8 # 0.3866667

## [1] 0.6111111

# specificity tnr--> specificity = tn/(tn+FP)  
tn8 <- taba4[1]  
tn\_plus\_fp8 <- taba4[1] + taba4[3]  
  
specificity8 <- tn8/tn\_plus\_fp8  
specificity8 # 0.9077381

## [1] 0.8955224

# 'Confusion matrix model with only important variable and new features'  
# 1 2  
# 1 305 31  
# 2 46 29  
  
'Model Performance'

## [1] "Model Performance"

#[1] "Accuracy" "81.26521"  
#[1] "sensitivity" "0.3866667"  
#[1] "specificity" "0.9077381"  
  
# We see that adding new feature has decreased accuracy.  
  
  
################################  
## Model Selection  
  
## There are many metrics which can be used to select the best model, choice of that metric is often done by cosidering its impact on a business KPI. For some business presision is important ,for others recall and in some cases overall acurracy might be important.  
  
Model <- data.frame(list("model\_name" = c("DT all variables","DT important variables","logistic all variables","logistic important variables","Logistic with feature engineering","RF with all variables","RF important variables"),  
 "Sensitivity" = c(sensitivity5,sensitivity6,sensitivitya,sensitivity4,sensitivity8,sensitivity7,sensitivity9),  
 "Specificity" = c(specificity5,specificity6,specificitya,specificity4,specificity8,specificity7,specificity9),  
 "Accuracy" = c(acc5,acc6,ac,acc4,acc8,acc7,acc9)))  
  
  
  
  
Model

## model\_name Sensitivity Specificity Accuracy  
## 1 DT all variables 0.7931034 0.8638743 85.88808  
## 2 DT important variables 0.3333333 0.9791667 86.13139  
## 3 logistic all variables 0.5600000 0.9017857 83.94161  
## 4 logistic important variables 0.5641026 0.9069069 84.18491  
## 5 Logistic with feature engineering 0.6111111 0.8955224 84.52088  
## 6 RF with all variables 0.4533333 0.9166667 83.21168  
## 7 RF important variables 0.3333333 0.9375000 82.72506

#################  
#From the above statistics we conclude that for this use case:  
   
 # - Decision Tree with all variables outperforms other models in terms of overall accuracy.

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

plot(roc(Atr\_Test$Attrition, Dt\_pre), print.auc=TRUE)

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

## Setting direction: controls < cases

plot(roc(Atr\_Test$Attrition, Dt\_pre1), print.auc = TRUE,col = "green", print.auc.y = .1, add = TRUE)

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

plot(roc(Atr\_Test$Attrition, Atr\_pre1), print.auc = TRUE,col = "blue", print.auc.y = .2, add = TRUE)

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

plot(roc(Atr\_Test$Attrition, Atr\_pre), print.auc = TRUE,col = "red", print.auc.y = .3, add = TRUE)

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

plot(roc(New\_Atr\_Test$Attrition, Atr\_pre4), print.auc = TRUE,col = "pink", print.auc.y = .4, add = TRUE)

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

