Employee Attrition

By -Riba Khan

2022-07-31

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Importing Data  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Employee\_Attrition <- read.csv("Attrition.csv")  
nrow(Employee\_Attrition)

## [1] 1470

ncol(Employee\_Attrition)

## [1] 35

#---------------Structure and summary of data----------------  
str(Employee\_Attrition)

## 'data.frame': 1470 obs. of 35 variables:  
## $ Age : int 41 49 37 33 27 32 59 30 38 36 ...  
## $ Attrition : chr "Yes" "No" "Yes" "No" ...  
## $ BusinessTravel : chr "Travel\_Rarely" "Travel\_Frequently" "Travel\_Rarely" "Travel\_Frequently" ...  
## $ DailyRate : int 1102 279 1373 1392 591 1005 1324 1358 216 1299 ...  
## $ Department : chr "Sales" "Research & Development" "Research & Development" "Research & Development" ...  
## $ 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 : chr "Life Sciences" "Life Sciences" "Other" "Life Sciences" ...  
## $ EmployeeCount : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ EmployeeNumber : int 1 2 4 5 7 8 10 11 12 13 ...  
## $ EnvironmentSatisfaction : int 2 3 4 4 1 4 3 4 4 3 ...  
## $ Gender : chr "Female" "Male" "Male" "Female" ...  
## $ 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 : chr "Sales Executive" "Research Scientist" "Laboratory Technician" "Research Scientist" ...  
## $ JobSatisfaction : int 4 2 3 3 2 4 1 3 3 3 ...  
## $ MaritalStatus : chr "Single" "Married" "Single" "Married" ...  
## $ 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 ...  
## $ Over18 : chr "Y" "Y" "Y" "Y" ...  
## $ OverTime : chr "Yes" "No" "Yes" "Yes" ...  
## $ 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(Employee\_Attrition)

## Age Attrition BusinessTravel DailyRate   
## Min. :18.00 Length:1470 Length:1470 Min. : 102.0   
## 1st Qu.:30.00 Class :character Class :character 1st Qu.: 465.0   
## Median :36.00 Mode :character Mode :character 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 EducationField   
## Length:1470 Min. : 1.000 Min. :1.000 Length:1470   
## Class :character 1st Qu.: 2.000 1st Qu.:2.000 Class :character   
## Mode :character Median : 7.000 Median :3.000 Mode :character   
## Mean : 9.193 Mean :2.913   
## 3rd Qu.:14.000 3rd Qu.:4.000   
## Max. :29.000 Max. :5.000   
## EmployeeCount EmployeeNumber EnvironmentSatisfaction Gender   
## Min. :1 Min. : 1.0 Min. :1.000 Length:1470   
## 1st Qu.:1 1st Qu.: 491.2 1st Qu.:2.000 Class :character   
## Median :1 Median :1020.5 Median :3.000 Mode :character   
## Mean :1 Mean :1024.9 Mean :2.722   
## 3rd Qu.:1 3rd Qu.:1555.8 3rd Qu.:4.000   
## Max. :1 Max. :2068.0 Max. :4.000   
## HourlyRate JobInvolvement JobLevel JobRole   
## Min. : 30.00 Min. :1.00 Min. :1.000 Length:1470   
## 1st Qu.: 48.00 1st Qu.:2.00 1st Qu.:1.000 Class :character   
## Median : 66.00 Median :3.00 Median :2.000 Mode :character   
## 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   
## JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate   
## Min. :1.000 Length:1470 Min. : 1009 Min. : 2094   
## 1st Qu.:2.000 Class :character 1st Qu.: 2911 1st Qu.: 8047   
## Median :3.000 Mode :character Median : 4919 Median :14236   
## Mean :2.729 Mean : 6503 Mean :14313   
## 3rd Qu.:4.000 3rd Qu.: 8379 3rd Qu.:20462   
## Max. :4.000 Max. :19999 Max. :26999   
## NumCompaniesWorked Over18 OverTime PercentSalaryHike  
## Min. :0.000 Length:1470 Length:1470 Min. :11.00   
## 1st Qu.:1.000 Class :character Class :character 1st Qu.:12.00   
## Median :2.000 Mode :character Mode :character Median :14.00   
## Mean :2.693 Mean :15.21   
## 3rd Qu.:4.000 3rd Qu.:18.00   
## Max. :9.000 Max. :25.00   
## PerformanceRating RelationshipSatisfaction StandardHours StockOptionLevel  
## Min. :3.000 Min. :1.000 Min. :80 Min. :0.0000   
## 1st Qu.:3.000 1st Qu.:2.000 1st Qu.:80 1st Qu.:0.0000   
## Median :3.000 Median :3.000 Median :80 Median :1.0000   
## Mean :3.154 Mean :2.712 Mean :80 Mean :0.7939   
## 3rd Qu.:3.000 3rd Qu.:4.000 3rd Qu.:80 3rd Qu.:1.0000   
## Max. :4.000 Max. :4.000 Max. :80 Max. :3.0000   
## TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany   
## Min. : 0.00 Min. :0.000 Min. :1.000 Min. : 0.000   
## 1st Qu.: 6.00 1st Qu.:2.000 1st Qu.:2.000 1st Qu.: 3.000   
## Median :10.00 Median :3.000 Median :3.000 Median : 5.000   
## Mean :11.28 Mean :2.799 Mean :2.761 Mean : 7.008   
## 3rd Qu.:15.00 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.: 9.000   
## Max. :40.00 Max. :6.000 Max. :4.000 Max. :40.000   
## YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager  
## Min. : 0.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.: 2.000 1st Qu.: 0.000 1st Qu.: 2.000   
## Median : 3.000 Median : 1.000 Median : 3.000   
## Mean : 4.229 Mean : 2.188 Mean : 4.123   
## 3rd Qu.: 7.000 3rd Qu.: 3.000 3rd Qu.: 7.000   
## Max. :18.000 Max. :15.000 Max. :17.000

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
##Installing necessary libraries  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
library(ggplot2) # For plotting  
library(class)  
library("pROC")

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

##   
## Attaching package: 'pROC'

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

library(rpart) # For Decision Tree  
library(rpart.plot)  
library(caret)

## Loading required package: lattice

library(nnet) # Forlogistic Regression  
library(randomForest) # For Random Forest

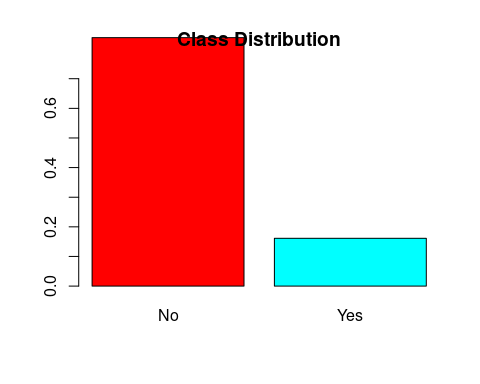
## randomForest 4.7-1.1

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

##   
## Attaching package: 'randomForest'

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

library(class) # For KNN calssifier  
library(e1071) # For naive bayes  
library("ROCR")  
  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Balancing The data  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
# Class imbalance check  
barplot(prop.table(table(Employee\_Attrition$Attrition)),  
 col = rainbow(2),  
 ylim = c(0, 0.7),  
 main = "Class Distribution")



# Observation : Attrition Yes > Attirition No  
# Data in not balanced so We will use UpSampling Techique  
  
#--------------UP Sampling---------------------#  
yes<- which(Employee\_Attrition$Attrition=="Yes")  
No<- which(Employee\_Attrition$Attrition=="No")  
length(yes)

## [1] 237

length(No)

## [1] 1233

Yes.upsam <- sample(yes, length(No), replace= T)  
Employee\_Attrition1<- Employee\_Attrition[c(No, Yes.upsam),]  
table(Employee\_Attrition1$Attrition)

##   
## No Yes   
## 1233 1233

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#importance of variables in dataset   
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
####usinng decision tree model to find out the importance###  
  
df<-rpart( Attrition~.,data=Employee\_Attrition1,control=rpart.control(minsplit = 10))  
df$variable.importance

## OverTime JobRole TotalWorkingYears   
## 149.1185480 90.4091879 87.1986316   
## MonthlyIncome YearsAtCompany Age   
## 60.9431696 32.2064100 30.5826958   
## NumCompaniesWorked YearsInCurrentRole JobLevel   
## 27.6066587 22.2679743 17.2957267   
## JobInvolvement EmployeeNumber MaritalStatus   
## 17.1026359 16.3554752 14.0360770   
## JobSatisfaction EducationField YearsWithCurrManager   
## 12.1006830 7.4826803 6.3687320   
## DailyRate HourlyRate PercentSalaryHike   
## 5.0120100 4.7816058 3.9839734   
## MonthlyRate TrainingTimesLastYear WorkLifeBalance   
## 3.8381760 3.4058774 2.7924653   
## YearsSinceLastPromotion Department Education   
## 2.1186531 1.8616435 0.7640068

#other way of doing it by Droping the the columns with no variability.  
drop\_var<-names(Employee\_Attrition1[, nearZeroVar(Employee\_Attrition1)])  
drop\_var #it will show the variable names with zero variability

## [1] "EmployeeCount" "Over18" "StandardHours"

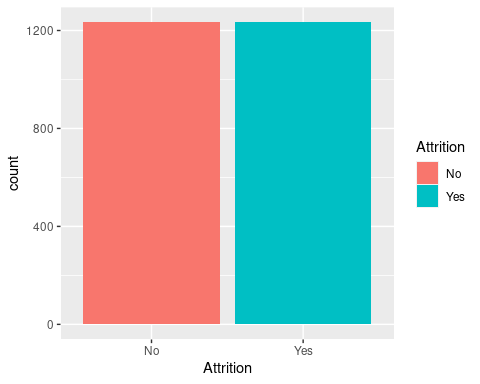
#Note:As we can notice EmployeeCount, Over18, Standhours has no variable importance  
  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## Cleaning the data  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
# checking null values in the dataset   
sum(is.na.data.frame(Employee\_Attrition1))

## [1] 0

# so there are no missing values which is good  
  
# Removing coloumns that are not needed for the analysis  
Employee\_Attrition1$Over18 <- NULL  
Employee\_Attrition1$EmployeeCount <- NULL  
Employee\_Attrition1$StandardHours <- NULL  
Employee\_Attrition1$EmployeeNumber<-NULL  
Employee\_Attrition1$DistanceFromHome<-NULL  
  
#------Check the dataset again--------  
#View(Employee\_Attrition1)  
# the coloumns have been removed  
  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Prparing the data for ML Algorithms  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   
# convert certain integer variable to factor variable as they make more sense as factor data type  
convert\_factor<- c("Education", "EnvironmentSatisfaction", "JobInvolvement",   
 "JobLevel","Department","EducationField","JobSatisfaction",   
 "PerformanceRating", "RelationshipSatisfaction",  
 "StockOptionLevel","TrainingTimesLastYear","WorkLifeBalance",  
 "BusinessTravel","Gender","JobRole","MaritalStatus","OverTime" )  
  
Employee\_Attrition1[, convert\_factor] <- lapply((Employee\_Attrition1  
 [, convert\_factor]), as.factor)  
Employee\_Attrition1[, convert\_factor] <- lapply((Employee\_Attrition1  
 [, convert\_factor]), unclass)  
  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Data Visulization   
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
table(Employee\_Attrition1$Attrition)

##   
## No Yes   
## 1233 1233

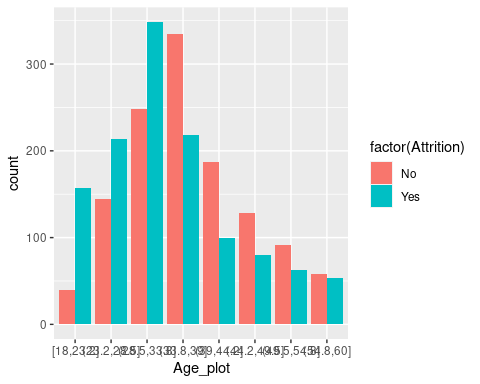
ggplot(Employee\_Attrition1, aes(x=Attrition, fill=Attrition)) + geom\_bar()



#-------Visulizing Factors having Higher Variability----------#  
  
#1) Age  
summary(Employee\_Attrition1$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 18.00 29.00 34.00 35.63 42.00 60.00

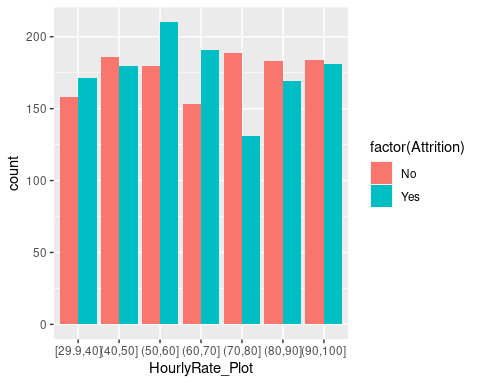
Age\_plot <- cut(Employee\_Attrition1$Age, 8, include.lowest = TRUE)  
ggplot(Employee\_Attrition1, aes(Age\_plot, ..count..,   
 fill = factor(Attrition))) + geom\_bar(position="dodge")



#2) Hourly Rate  
summary(Employee\_Attrition1$HourlyRate)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 30.00 48.00 65.00 65.52 84.00 100.00

HourlyRate\_Plot<- cut(Employee\_Attrition1$HourlyRate, 7, include.lowest = TRUE)  
ggplot(Employee\_Attrition1, aes(HourlyRate\_Plot, ..count..,  
 fill = factor(Attrition))) + geom\_bar(position="dodge")



# 3) Job Role  
#table(Employee\_Attrition1$JobRole, Employee\_Attrition1$Attrition)  
ggplot(Employee\_Attrition1, aes(JobRole, ..count..,   
 fill = factor(Attrition))) + geom\_bar(position="dodge")

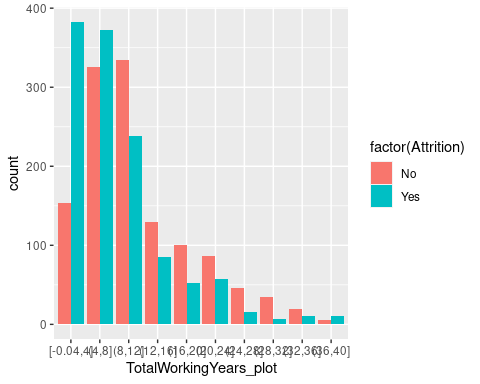
**Chart, bar chart

Description automatically generated**

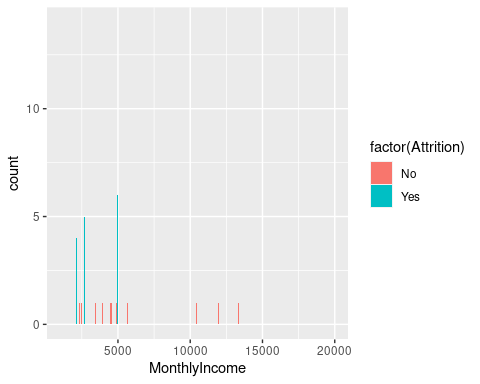
# 4) Total working years  
summary(Employee\_Attrition1$TotalWorkingYears)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 5.00 8.50 10.19 13.00 40.00

TotalWorkingYears\_plot <- cut(Employee\_Attrition1$TotalWorkingYears,  
 10, include.lowest = TRUE)  
ggplot(Employee\_Attrition1, aes(TotalWorkingYears\_plot, ..count..,   
 fill = factor(Attrition))) + geom\_bar(position="dodge")



# 5) Monlthy Income  
#table(Employee\_Attrition1$MonthlyIncome, Employee\_Attrition1$Attrition)  
ggplot(Employee\_Attrition1, aes(MonthlyIncome, ..count..,   
 fill = factor(Attrition))) + geom\_bar(position="dodge")

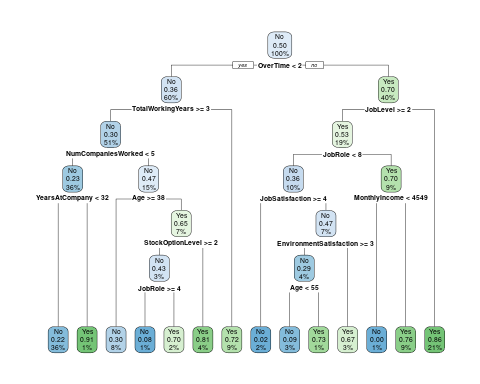


# 6) Overtime  
#table(Employee\_Attrition1$OverTime, Employee\_Attrition1$Attrition)  
ggplot(Employee\_Attrition1, aes(OverTime, ..count..,   
 fill = factor(Attrition))) + geom\_bar(position="dodge")

**Graphical user interface

Description automatically generated**

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Data partition  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Dividing the data into test and train  
p <- 0.30  
set.seed(123)  
ind<- sample(c(rep(0, (1-p) \* nrow(Employee\_Attrition1)), rep(1, p \* nrow(Employee\_Attrition1))))  
X1 <-Employee\_Attrition1[ind==0,] #Training  
X2 <- Employee\_Attrition1[ind==1,] #Test  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Machine learning Models  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
#=======================  
# 1) Decision Tree Model  
#=======================  
  
DT\_model<- rpart(Attrition~., data = X1, method = 'class')  
rpart.plot(DT\_model, extra = 106)



# Prediction  
ypred <- predict(DT\_model, newdata= X2, type ="class")  
  
#Confusion Matrix of Decision Tree  
T <- table(X2$Attrition, ypred)  
#Prediction Accuracy  
acu<- sum(diag(T))/ sum(T) \*100  
  
#Prediction Error  
Pre <- 100- acu  
  
# Percision  
pe <- T[1,1]/sum(T[,1])  
  
#Recall  
Re <- T[1,1]/sum(T[1,])  
  
cat("For Desicion Tree","\n", "Accuracy =", acu,"\n", "Prediction Error=", Pre,  
"\n", "Percision=", pe ,"\n", "Recall= ", Re, "\n")

## For Desicion Tree   
## Accuracy = 77.7027   
## Prediction Error= 22.2973   
## Percision= 0.7869318   
## Recall= 0.7547684

T

## ypred  
## No Yes  
## No 277 90  
## Yes 75 298

#Predict the probability For test data  
Prob\_DT <- predict(DT\_model, newdata= X2[,-2], type ="prob")  
  
#AUC value  
AUC <- auc(X2$Attrition,Prob\_DT[,2])

## Setting levels: control = No, case = Yes

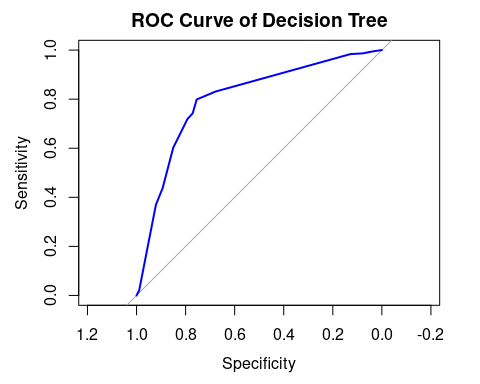
## Setting direction: controls < cases

AUC

## Area under the curve: 0.802

# Roc curve  
plot(roc(X2$Attrition,Prob\_DT[,2]), main="ROC Curve of Decision Tree",col="blue")

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases



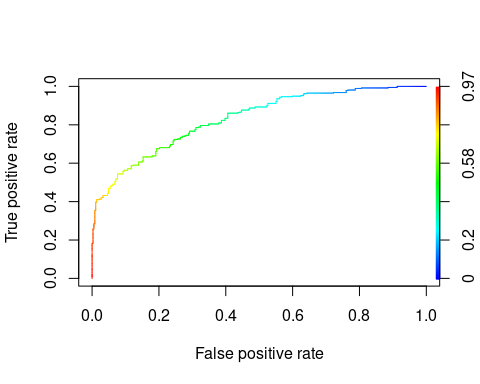
#=======================  
# 2) Logistic Regression  
#=======================  
  
set.seed(123)  
Logistic\_Model <- glm(as.factor(Attrition)~., data=X1, family = 'binomial')  
pred <- predict(Logistic\_Model, newdata = X2, type="response")  
  
#Predictions  
test <- vector()  
test[pred > mean(Logistic\_Model$y==1)] <- "Yes"  
test[pred<= mean(Logistic\_Model$y==0)] <- "No"  
  
#Confusion matrix  
T1 <- table(test, X2[,2])  
  
#Prediction Accuracy  
acu1 <- 100\*sum(diag(T1))/sum(T1)  
  
# Prediction Error  
Pre1 <- 100-acu1  
  
# Percision  
pe1 <- T1[1,1]/sum(T1[,1])  
  
#Recall  
Re1 <- T1[1,1]/sum(T1[1,])  
  
cat("For Logistic Regression","\n", "Accuracy =", acu1,"\n", "Prediction Error=", Pre1,  
"\n", "Percision=", pe1 ,"\n", "Recall= ", Re1, "\n")

## For Logistic Regression   
## Accuracy = 73.78378   
## Prediction Error= 26.21622   
## Percision= 0.7520436   
## Recall= 0.7282322

T1

##   
## test No Yes  
## No 276 103  
## Yes 91 270

#ROC and AUC  
pred\_resp <- predict(Logistic\_Model ,X2,type="response")  
pred <- prediction(pred\_resp, X2[,2])  
perf <- performance(pred, "tpr", "fpr")  
plot(perf, colorize=TRUE)



#  
AUC1<-auc(X2[,2], pred\_resp)

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

#==================  
# 3) Random Forest  
#==================  
  
model\_RF <- randomForest(x= X1[,-2], y=as.factor(X1[,2]),  
 ntree=500,  
 mtry=2,  
 importance=TRUE,  
 replace=FALSE)  
  
# Predicting the Test set results  
y\_pred = predict(model\_RF, newdata = X2[-2])  
  
  
#Confusion matrix  
T2 <- table(y\_pred, X2[,2])  
  
#Prediction Accuracy  
acu2 <- 100\*sum(diag(T2))/sum(T2)  
  
# Prediction Error  
Pre2 <- 100-acu2  
  
# Percision  
pe2 <- T2[1,1]/sum(T2[,1])  
  
#Recall  
Re2 <- T2[1,1]/sum(T2[1,])  
  
  
cat("For Random forest","\n", "Accuracy =", acu2,"\n", "Prediction Error=", Pre2,  
"\n", "Percision=", pe2 ,"\n", "Recall= ", Re2, "\n")

## For Random forest   
## Accuracy = 97.02703   
## Prediction Error= 2.972973   
## Percision= 0.9618529   
## Recall= 0.9778393

T2

##   
## y\_pred No Yes  
## No 353 8  
## Yes 14 365

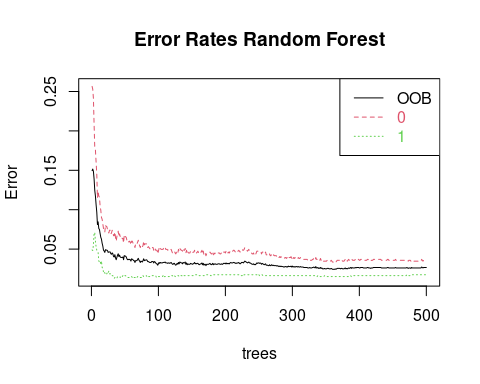
# Generate a textual view of the Random Forest  
model\_RF

##   
## Call:  
## randomForest(x = X1[, -2], y = as.factor(X1[, 2]), ntree = 500, mtry = 2, replace = FALSE, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 2.67%  
## Confusion matrix:  
## No Yes class.error  
## No 835 31 0.03579677  
## Yes 15 845 0.01744186

summary(model\_RF)

## Length Class Mode   
## call 7 -none- call   
## type 1 -none- character  
## predicted 1726 factor numeric   
## err.rate 1500 -none- numeric   
## confusion 6 -none- numeric   
## votes 3452 matrix numeric   
## oob.times 1726 -none- numeric   
## classes 2 -none- character  
## importance 116 -none- numeric   
## importanceSD 87 -none- numeric   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 14 -none- list   
## y 1726 factor numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL

plot(model\_RF, main="")  
legend("topright", c("OOB", "0", "1"), text.col=1:6, lty=1:3, col=1:3)  
title(main="Error Rates Random Forest")



# ROC and AUC  
  
Prob\_RF <- (predict(model\_RF, X2, type = 'prob'))  
#AUC value  
AUC2 <- auc(X2$Attrition,Prob\_RF[,2])

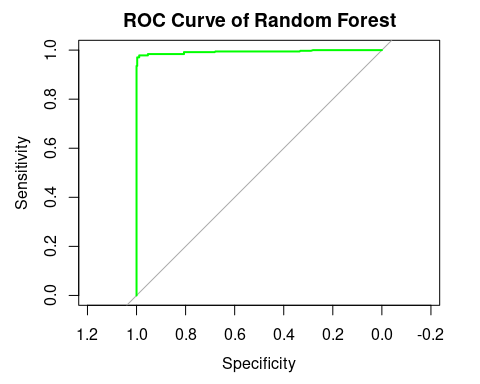
## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

AUC2

## Area under the curve: 0.9935

# Roc curve  
plot(roc(X2$Attrition,Prob\_RF[,2]), main="ROC Curve of Random Forest",col="Green")

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases



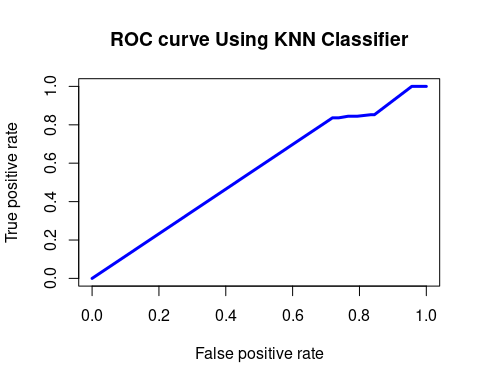
#=============  
#4) KNN  
#=============  
  
  
x1<- X1[,-2]  
x2<- X2[,-2]  
  
KN<-knn(train=x1, test= x2, cl=X1[,2], k = 3)  
  
  
#Confusion matrix  
T3 <- table( KN, X2[,2])  
  
#Prediction Accuracy  
acu3 <- 100\*sum(diag(T3))/sum(T3)  
  
# Prediction Error  
Pre3 <- 100-acu3  
  
# Percision  
pe3 <- T3[1,1]/sum(T3[,1])  
  
#Recall  
Re3 <- T3[1,1]/sum(T3[1,])  
  
  
cat("For KNN Classifier","\n", "Accuracy =", acu3,"\n", "Prediction Error=", Pre3,  
"\n", "Percision=", pe3 ,"\n", "Recall= ", Re3, "\n")

## For KNN Classifier   
## Accuracy = 77.83784   
## Prediction Error= 22.16216   
## Percision= 0.599455   
## Recall= 0.92827

T3

##   
## KN No Yes  
## No 220 17  
## Yes 147 356

#AUC and ROC  
  
KN\_prob<-class::knn(train=x1, test= x2, cl=X1[,2], k = 3, prob= TRUE)  
  
prob <- attr(KN\_prob, "prob")  
prob <- 2\*ifelse(KN\_prob == "-1", 1-prob, prob) - 1  
  
pred\_knn <- prediction(prob, X2[,2])  
pred\_knn <- performance(pred\_knn, "tpr", "fpr")  
plot(pred\_knn, col="blue", lwd=3, main="ROC curve Using KNN Classifier")



AUC3<-auc(X2[,2], prob)

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

#==============  
#5)Naive Bayes  
#==============  
  
classifier\_cl <- naiveBayes( X1[,2]~ ., data = X1)  
  
   
# Predicting on test data'  
y\_pred <- predict(classifier\_cl, newdata = X2[,-2])  
  
#Confusion matrix  
T4 <- table( y\_pred, X2[,2])  
  
#Prediction Accuracy  
acu4 <- 100\*sum(diag(T4))/sum(T4)  
  
# Prediction Error  
Pre4 <- 100-acu4  
  
# Percision  
pe4 <- T4[1,1]/sum(T4[,1])  
  
#Recall  
Re4 <- T4[1,1]/sum(T4[1,])  
  
  
cat("For Naive bayes","\n", "Accuracy =", acu4,"\n", "Prediction Error=", Pre4,  
"\n", "Percision=", pe4 ,"\n", "Recall= ", Re4, "\n")

## For Naive bayes   
## Accuracy = 64.86486   
## Prediction Error= 35.13514   
## Percision= 0.5558583   
## Recall= 0.6777409

T4

##   
## y\_pred No Yes  
## No 204 97  
## Yes 163 276

#AUC and ROC  
  
y\_pred\_prob <- predict(classifier\_cl, newdata = X2[,-2], type = 'raw')  
#AUC value  
AUC4 <- auc(X2$Attrition,y\_pred\_prob[,2])

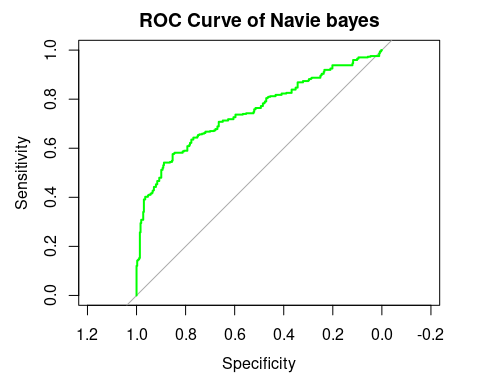
## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

AUC4

## Area under the curve: 0.749

# Roc curve  
plot(roc(X2$Attrition,y\_pred\_prob[,2]), main="ROC Curve of Navie bayes",col="green")

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases



# combain Roc Curve  
par(mfrow=c(2,3))  
plot(roc(X2$Attrition,Prob\_DT[,2]), main="ROC Curve of Decision Tree",col="blue")

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

plot(perf, colorize=TRUE, main= "ROC Curve for Logistic regression")  
plot(roc(X2$Attrition,Prob\_RF[,2]), main="ROC Curve of Random Forest",col="Green")

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

plot(pred\_knn, col="blue", lwd=3, main="ROC curve Using KNN Classifier")  
plot(roc(X2$Attrition,y\_pred\_prob[,2]), main="ROC Curve of Navie bayes",col="green")

## Setting levels: control = No, case = Yes  
## Setting direction: controls < cases

# Combine table  
acuracy <- c(acu,acu1,acu2,acu3,acu4)  
Prediction\_error<- c(Pre,Pre1, Pre2, Pre3, Pre4)  
Percision <- c(pe, pe1, pe2, pe3, pe4)  
Recall <- c(Re, Re1, Re2, Re3, Re4)  
AUCC <- c(AUC, AUC1, AUC2, AUC3, AUC4)  
c.table <- cbind(acuracy, Prediction\_error ,Percision,Recall ,AUCC)  
rownames(c.table)<- c("Decision Tree", "Logistic Regression", "Random Forest","KNN","Naive bayes")  
  
  
# From the table It can be observed that accuracy of Random forest is Highest,  
#Prediction\_error is lowest and other measures are also good so we use Random forest  
#for further predictions  
  
  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#Final Predictions On real data  
#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
y\_predF = predict(model\_RF, newdata = Employee\_Attrition[-2])  
  
  
#Confusion matrix  
Tf <- table(y\_predF, Employee\_Attrition[,2])  
  
#Prediction Accuracy  
acuF <- 100\*sum(diag(Tf))/sum(Tf)  
  
# Prediction Error  
Pref <- 100-acuF  
  
# Percision  
peF <- Tf[1,1]/sum(Tf[,1])  
  
#Recall  
Ref <- Tf[1,1]/sum(Tf[1,])  
  
  
cat("Final Predictions","\n", "Accuracy =", acuF,"\n", "Prediction Error=", Pref,  
"\n", "Percision=", peF ,"\n", "Recall= ", Ref, "\n")

## Final Predictions   
## Accuracy = 98.36735   
## Prediction Error= 1.632653   
## Percision= 0.9886456   
## Recall= 0.9918633

Tf

##   
## y\_predF No Yes  
## No 1219 10  
## Yes 14 227

