#Load the Data into R

hrdata = read.csv("HR\_data.csv")

head(hrdata)

summary(hrdata)

#Correlation of Data

colnames(hrdata)

cordata = cor(hrdata[,1:8])

library(corrplot)

corrplot(cordata,type = "full")

#Visualizing the characteristics of people who left

leftdata = hrdata[hrdata$left==1,]

par(mfrow = c(2,3))

hist(leftdata$satisfaction\_level,col = "#3090C7",main = "", xlab = "Satisfaction level")

hist(leftdata$Work\_accident,col = "#3090C7",main = "", xlab = "Work Accidents")

hist(leftdata$average\_montly\_hours,col = "#3090C7",main = "", xlab = "AVG monthly hours")

hist(leftdata$last\_evaluation,col = "#3090C7",main = "", xlab = "Last Evaluation scores")

hist(leftdata$number\_project,col = "#3090C7",main = "", xlab = "No.of Projects")

plot(as.factor(leftdata$promotion\_last\_5years),col = "#3090C7",main = "", xlab = "Promotion last 5 years")

#Plotting a visualiztion for people with average working hrs vs number of projects, with coloring according to left or not

par(mfrow = c(1,1))

plot(hrdata$number\_project,hrdata$average\_montly\_hours,col=as.factor(hrdata$left))

#Lets calculate daily working hour for the employees who left the job

leftdata$average\_daily\_working\_hours<-(leftdata$average\_montly\_hours/30)

hist(leftdata$average\_daily\_working\_hours)

#Visualizing how many people who have left have met a work accident,Since it's a binary variable convert to factor

leftdata$Work\_accident<-as.factor(leftdata$Work\_accident)

plot(leftdata$Work\_accident)

#Visualizing how many people who have left have got a promotion in last five years,Since it's a binary variable convert to factor

leftdata$promotion\_last\_5years<-as.factor(leftdata$promotion\_last\_5years)

plot(leftdata$promotion\_last\_5years)

#Plotting satisfaction level of people who left the job vs who didn't

set.seed(42)

p1 <- hist(hrdata$satisfaction\_level[hrdata$left=="1"])

p2 <- hist(hrdata$satisfaction\_level[hrdata$left=="0"])

plot( p2, col=rgb(0,0,1,1/4), xlim=c(0,1)) # first histogram

plot( p1, col=rgb(1,0,0,1/4), xlim=c(0,1), add=T) # second

##Let's see department wise turn out

dat = aggregate(left~department,data = leftdata,FUN = sum)

View(dat)

plot(table(leftdata$department))

#The above graph doesnot take into account the number of employees there in each department

plot(table(leftdata$department)/table(hrdata$department),las =2)

#People who should'nt have left

priorityData = leftdata[leftdata$satisfaction\_level>0.7|leftdata$last\_evaluation>0.8|leftdata$number\_project>5,]

nrow(priorityData)

#Lets check the correlation of people who have already left

corleft = cor(priorityData[,c(1,2,3,4,5,6,8)])

corrplot(corleft,type = "full")

#Now we will predict what are the factors that causes them to leave

hrdata$left = ifelse(hrdata$left==1,"Left","Not Left")

hrdata$left = as.factor(hrdata$left)

library(caTools)

id = sample.split(hrdata,SplitRatio = 0.7)

#id = sample(2,nrow(hrdata), prob = c(0.8,0.2),replace = T)

traindata = hrdata[id,]

testdata = hrdata[!id,]

#Decision Tree\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

library(rpart)

treeModel = rpart(left~.,data = traindata)

treeModel

plot(treeModel,margin = 0.2)

text(treeModel,cex = 0.7)

predtree = predict(treeModel,testdata,type = "class")

predtree

library(caret)

confusionMatrix(table(predtree,testdata$left))

#RandomForest\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

library(randomForest)

forestModel = randomForest(left~.,data = traindata)

forestModel

predforest = predict(forestModel,testdata,type = "class")

confusionMatrix(table(predforest,testdata$left))

#Naive-Bayes\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

library(e1071)

bayesModel = naiveBayes(left~.,data = traindata)

bayesModel

predBayes = predict(bayesModel,testdata,type = "class")

confusionMatrix(table(predBayes,testdata$left))

#SVM\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

svmModel = svm(left~.,data = traindata)

svmModel

predsvm = predict(svmModel,testdata)

confusionMatrix(table(predsvm,testdata$left))