```
#setting working directory
setwd("C:/Users/Sarthak Gupta/Desktop/DATA SCIENTIST/EDWISOR/PROJECT/PROJECT 2")

#checking the working directory has been set up
getwd()

#importing the dataset
data1 <- read.csv("C:/Users/Sarthak Gupta/Desktop/DATA SCIENTIST/EDWISOR/PROJECT/PROJECT 2/train_data.csv",header = T)
test<-read.csv("C:/Users/Sarthak Gupta/Desktop/DATA SCIENTIST/EDWISOR/PROJECT/PROJECT 2/test_data.csv",header = T)

#Exploratory Data Analysis
View(data1)
head(data1)
tail(data1)
str(data1)
table(data1$Churn)
summary(data1)


#Missing Value Analysis
is.na(data1)
table(is.na(data1))
is.na(test)
table(is.na(test))
#No missing value found


# These variables are not part of predictor variables as per problem statement.
data2<- data1[,-c(1,3,4)]

test1<- test[,-c(1,3,4)]

#outlier analysis
numeric_index = sapply(data2,is.numeric) #selecting only numeric

numeric_data = data2[,numeric_index]

cnames = colnames(numeric_data)

#Delete the outliers using boxplot method
for(i in cnames){
     print(i)
     val = data2[,i][data2[,i] %in% boxplot.stats(data2[,i])$out]
     #print(length(val))
     data2 = data2[which(!data2[,i] %in% val),]
}

#Feature Selection
## Correlation Plot (for numeric varaiables)

library(corrgram)
corrgram(data2[,numeric_index])

#Deleting variables having high corelation with another independent variable
data3 = subset(data2,select = -c(total.day.charge,total.eve.charge,total.night.charge,total.intl.charge))

test1 = subset(test1,select = -c(total.day.charge,total.eve.charge,total.night.charge,total.intl.charge))
## Chi-squared Test of Independence
factor_index = sapply(data3,is.factor)
factor_data = data3[,factor_index]

for (i in 1:2)
{
  print(names(factor_data)[i])
  print(chisq.test(table(factor_data$Churn,factor_data[,i])))
}




data4 = data3

############Model Building and Prediction

#Decision Tree

library(C50)
c50_model = C5.0(Churn~.,data4,trials=100,rules=TRUE)

#summary of DT model
summary(c50_model)

#write rules into disk
write(capture.output(summary(c50_model)),"c50Rules.txt")
```

```
#Prediction on test data
c50_Predictions = predict(c50_model,test1[,-14],type = "class")

#Evaluate the performance of the model
ConfMatrix_c50 = table(test1$Churn,c50_Predictions)
library(caret)
confusionMatrix(ConfMatrix_c50)
#Accuracy is 93.88
submission_DT_R = cbind(test,Prediction = c50_Predictions)
#write a csv output
write.csv(submission_DT_R,"submission_DT_R.csv",row.names = F)

###############

#Randomforest
library(randomForest)
RF_model = randomForest(Churn~.,data4,importance = TRUE,ntree = 100)

#Extract rules from random forest
#Transform rf object to an inTrees' Format
library(inTrees)
treelist = RF2List(RF_model)

#Extract Rules
exec = extractRules(treelist,data4[,-14])

#Visualize some rules
exec[1:2,]


#Make rules more readable

readableRules = presentRules(exec,colnames(data4))

readableRules[1:2,]

#Get rule metrics
ruleMetric = getRuleMetric(exec,data4[,-14],data4$Churn)

#Evaluate few rules
ruleMetric[1:2,]

#Predict test data using random forest model
RF_Predictions = predict(RF_model,test1[,-14])

#Evaluate the performance
confMatrix_RF = table(test1$Churn,RF_Predictions)
confusionMatrix(confMatrix_RF)
#Accuracy is 92.8%
submission_RandomForest_R = cbind(test,Prediction = RF_Predictions)
#write a csv output
write.csv(submission_RandomForest_R,"submission_RandomForest_R.csv",row.names = F)

##################
#Logistic Regression
logit_model = glm(Churn~.,data=data4,family = "binomial")

#summary of model
summary(logit_model)

#predict usng logistic regression
logit_Predictions = predict(logit_model,newdata = test1,type = "response")

#convert into probabilities
logit_Predictions = ifelse(logit_Predictions>0.5,1,0)

#Evaluate the performance
confMatrix = table(test1$Churn,logit_Predictions)

#Accuracy = (1424+60)*100/(1424+60+19+164) = 89.02%

submission_LogisticRegression_R = cbind(test,Prediction = logit_Predictions)
submission_LogisticRegression_R$Prediction = ifelse(submission_LogisticRegression_R$Prediction ==0 ,"False.","True.")
#write a csv output
write.csv(submission_LogisticRegression_R,"submission_LogisticRegression_R.csv",row.names = F)

######################

#NaiveBayes
library(e1071)

NB_model = naiveBayes(Churn~.,data =data4)

#prediction
```

```
NB_Prediction = predict(NB_model,test1[,1:13],type = 'class')

#confusion matrix
Conf_matrix = table(observed = test1[,14],predicted = NB_Prediction)
confusionMatrix(Conf_matrix)
#Accuracy is 88.84%

submission_NaiveBayes_R = cbind(test,Prediction = NB_Prediction)
#write a csv output
write.csv(submission_NaiveBayes_R,"submission_NaiveBayes_R.csv",row.names = F)
```