My Code:

#libraries

library(readxl)

library(plyr)

library(e1071)

library(rpart)

library(randomForest)

library(dplyr)

library(ggplot2)

setwd("/Users/rushikeshkhankar/Desktop/R")

getwd()

#Loading Dataset

data <- read.csv("~/Desktop/R/Project/Projects for Submission/Education/Project 1\_Dataset.csv")

View(data)

#Data Sanity Check

str(data)

#Checking Null values in Dataset

is.null(data)

#Descriptive analysis

summary(data)

str(data)

boxplot(data$gre)

quantile(data$gre, c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,0.97,0.98,0.985,0.99,0.995,1))

data<-data[data$gre>=350,]

boxplot(data$gpa)

quantile(data$gpa, c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,0.97,0.98,0.985,0.99,0.995,1))

data<-data[data$gpa>=2.5,]

#Logistic REgression on full data -- 70.10% accuracy

logistic <- glm(admit ~ ., data = data, family = binomial())

logistic

#Pedictef Probabilities

result <- predict(logistic, data)

result

summary(result)

res <- ifelse(result > 0,1,0)

#Accuracy of the model

accuracy <- table(res, data[,1])

sum(diag(accuracy))/sum(accuracy)

#Dropping the insignificant variables

data <- data[, -c(4,5,6)]

data1 <- data

#Converting necessary variables into factor

data$admit <- as.factor(data$admit)

data$admit

#To decide the which model more sutaible for Dataset I compare Naive bayes, SVM, Decision Tree and Random Forest

#Naive Bayes----

naive\_bayes <- naiveBayes(admit~., data = data)

summary(naive\_bayes)

Predictions <- predict(naive\_bayes, data)

Predictions

table(Predictions, data$admit)

#accuracy of model

table\_mat <- table(Predictions, data$admit)

accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

accuracy\_Test

#SVM

SVM <- svm(admit ~., data = data, method = 'class')

summary(SVM)

Predictions <- predict(SVM, data, type ='class')

Predictions

table(Predictions, data$admit)

#Accuracy of model

table\_mat <- table(Predictions, data$admit)

accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

accuracy\_Test

#Decision Tree

tree <- rpart(admit~., data = data, method = 'class')

summary(tree)

Predictions <- predict(tree, data, type = 'class')

Predictions

table(Predictions, data$admit)

#Accuracy of model

table\_mat <- table(Predictions, data$admit)

accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

accuracy\_Test

#Random Forest

forest <- randomForest(x = data, y = data$admit, ntree = 800)

summary(forest)

Predictions <- predict(forest, data)

Predictions

table(Predictions, data$admit)

#Accuracy of model

table\_mat <- table(Predictions, data$admit)

accuracy\_Test <- sum(diag(table\_mat))/sum(table\_mat)

accuracy\_Test

#Categorize the grade point average into high or mediumand plot it on point chart

Aptitude\_Descriptive = transform(data1,

GreLevels = ifelse(gre<439, "Low", ifelse(gre<579,"Medium","High")))

str(Aptitude\_Descriptive)

Sum\_Apt = aggregate(admit~GreLevels, data = Aptitude\_Descriptive, FUN = sum)

lenght\_Apt = aggregate(admit~GreLevels, Aptitude\_Descriptive, FUN = length)

Probability\_Table = cbind(Sum\_Apt, Recs = lenght\_Apt[,2])

Probability\_Table\_Final = transform(Probability\_Table, Probability\_Admission = admit/Recs)

Probability\_Table\_Final

ggplot(Probability\_Table\_Final, aes(x=GreLevels, y=Probability\_Admission))+geom\_point()

#Cross grid for admission variable with GRE categorized

table(Aptitude\_Descriptive$admit, Aptitude\_Descriptive$GreLevels)