```
#######SOURCE CODE of Project5 - College Admission


setwd("C://Users//sinun//OneDrive//Documents//Simplilearn//R
programming//project5")

data<- read.csv("College_admission.csv")

str(data)
summary(data)
head(data)
data_org=data

#Find the missing values. (if any, then perform outlier treatment)
sapply(data, function(x) sum(is.na(x)))
data <- na.omit(data)
#No missing values

# Find outliers (if any, then perform outlier treatment)
boxplot(data$gre)
quantile(data$gre,c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,0.99,0.995,1))

data2 <- data[data$gre >399, ]
boxplot(data2$gre)

boxplot(data2$gpa)
quantile(data2$gpa,c(0,0.05,0.1,0.25,0.5,0.75,0.90,0.95,0.99,0.995,1))

data3 <- data2[data2$gpa >2.2600399, ]
boxplot(data3$gpa)
nrow(data)-nrow(data3)
data <- data3

#Find the structure of the data set
str(data)
sapply(data, class)
nrow(data)
names(data)

# Find whether the data is normally distributed or not. Use the plot to
determine the same.
plot(data)
hist(data$gre,col="Red", main="Graduate record exam score")
hist(data$gpa,col="Yellow", main="Grade point average")

# Normalize the data if not normally distributed.
data$gre<-scale(data$gre,center = TRUE,scale=TRUE)
data$gpa<-scale(data$gpa,center = TRUE,scale=TRUE)

hist(data$gre,col="Red", main="Graduate record exam score")
hist(data$gpa,col="Yellow", main="Grade point average")

#Use variable reduction techniques to identify significant variables.
#install.packages("caret")
```

```
library(caret)

regressor <- lm(admit ~ ., data= data)
summary(regressor)
# gre,gpa and rank are having relatively less p value and they can be considered
as Significant variables

# Run logistic model to determine the factors that influence the admission
process of a student (Drop insignificant variables)
library(caTools)
#splitting dataset into train and test
set.seed(0)
split<-sample.split(data, SplitRatio=.75)
train<-subset(data,split==T)
test<-subset(data,split==F)

##Logistic regression model 1
logic_reg1<-glm(admit~.,data=train,family='binomial')
summary(logic_reg1)

##Logistic regression model 2
logic_reg2<-glm(admit~gre+gpa+rank,data=train,family='binomial')
summary(logic_reg2)

#Final model of logistic model
logic_reg3<-glm(admit~gpa+rank,data=train,family='binomial')
summary(logic_reg3)

predict_Test<-predict(logic_reg3,test,type="response")
#predict_Test

predict_Test<-ifelse(predict_Test>0.5,1,0)
#Calculate the accuracy of the model and run validation techniques.
confusion_matrix=table(actual=test$admit,predicted=predict_Test)
confusion_matrix
missing_classerr<-mean(predict_Test!=test$admit)
print(paste('Accuracy=',1-missing_classerr))
#Accuracy of Logistic regression model= 70.37 %

# Validation Techniques

#VIF means detecting presence of multicollinearity
library(car)
vif(logic_reg3)
 #vif<2, so model is good

#serial Correlation or Autocorrelation
library("lmtest")

#Durbin watson Test
dwtest(logic_reg3)
#p value greater than 0.05, so model is good
```

```
# Try other modelling techniques like decision tree and SVM and select a
champion model

#SVM model
library(e1071)
svm_clf=svm(admit~gre+gpa+rank,data=train,type
='C-classification',kernel='linear')
summary(svm_clf)
predicted_val2<-predict(svm_clf,test)
#predicted_val2

confusion_matrix2=table(actual=test$admit,predicted=predicted_val2)
confusion_matrix2

missing_classerr2<-mean(predicted_val2!=test$admit)
print(paste('Accuracy=',1-missing_classerr2))

#Accuracy of SVM model= 65.74 %

#Decision tree
library(rpart)
#install.packages("rpart.plot")
library(rpart.plot)
v <- data$admit
table(v)
set.seed(522)
# runif function returns a uniform distribution which can be further
conditionally split into 75-25 ratio
data[, 'train'] <- ifelse(runif(nrow(data)) < 0.75, 1, 0)
nrow(data)
#data[, 'train']

trainSet <- data[data$train == 1,]
testSet <- data[data$train == 0, ]

trainColNum <- grep('train', names(trainSet))

trainSet <- trainSet[, -trainColNum]
testSet <- testSet[, -trainColNum]

treeFit <- rpart(admit~gre+gpa+rank,data=trainSet,method = 'class')
print(treeFit)

rpart.plot(treeFit, box.col=c("red", "green"))

Prediction1 <- predict(treeFit,newdata=testSet[-5],type = 'class')


cm <- table(testSet$admit, Prediction1)
cm
misClassError <- mean(Prediction1 != testSet$admit)
print(paste('Accuracy =', 1-misClassError))

# accuracy of Decision tree based Algorith= 64.70 %
```

```
## Naive Bayes Technique
library(e1071)
library('caTools')
nb<-naiveBayes(admit~gpa+rank+gre,data=train)
nb
predicted_val5<-predict(nb, test, type="class")
misclassError1<-mean(predicted_val5!=test$admit)
print(paste('Accuracy=',1-misclassError1))
##"Accuracy for Naive Bayes Technique= 71.29 %"

#Accuracy of Logistic regression model   = 70.37 %
#Accuracy of SVM model                   = 65.74 %
#Accuracy of DEcision tree model         = 64.70 %
#Accuracy for Naive Bayes Technique      = 71.29 %

#Among the models used, Logistic regression model and Naive Bayes models are
champion models


#install.packages("ggplot2")
library(ggplot2)

#Categorize the grade point average into High, Medium, and Low (with admission
probability percentages) and plot it on a point chart.
# from summary we can understand that min of gpa is 2.26 and max is 4
Descriptive=transform(data_org,GPA_Levels=ifelse(data_org$gpa<3,"Low",ifelse(dat
a_org$gpa>=3&data_org$gpa<=3.49,"Medium","High")))
View(Descriptive)
Sum_Desc=aggregate(admit~GPA_Levels,Descriptive,FUN=sum)
Sum_Desc
# From output its observed that :
# Total no of students admitted= 127
# Total no of students applied = 400
# probability of a student get admitted = 127/400= 31.75 %
length_Desc=aggregate(admit~GPA_Levels,Descriptive,FUN=length)
Probability_Table=cbind(Sum_Desc,total_applicants=length_Desc[,2])
Probability_Table
Probability_Table_final=transform(Probability_Table,Probability_Admission=(admit
/127))
ggplot(Probability_Table_final,aes(x=GPA_Levels,y=Probability_Admission))+geom_p
oint()

#No of students having High GPA(from 3.5 to 4.0)admitted =68
#Probability of students with High GPA getting admitted= 68/127= 53.54 %

#No of students having  Medium GPA(from 3.0 to 3.49)admitted=44
#Probability of students with Medium GPA getting admitted= 44/127=34.647%

#No of students having  Low GPA(from 2.0 to 2.9)admitted=15
##Probability of students with Low GPA getting admitted=15/127= 11.817%

data_org$GPA_levels=ifelse(data_org$gpa<3,"Low",ifelse(data_org$gpa>=3&data_org$
```

```
gpa<=3.49,"Medium","High"))

#cross grid for admission variable with GRE categorized
Descriptive2=transform(data_org,GreLevels=ifelse(data_org$gre<440,"Low",ifelse(data_org$gre<580,"Medium","High")))
View(Descriptive2)
Sum_Desc2=aggregate(admit~GreLevels,Descriptive2,FUN=sum)
Sum_Desc2
length_Desc2=aggregate(admit~GreLevels,Descriptive2,FUN=length)
Probability_Table2=cbind(Sum_Desc2,Recs=length_Desc2[,2])
Probability_Table_final2=transform(Probability_Table2,Probability_Admission2=admit/127)
ggplot(Probability_Table_final2,aes(x=GreLevels,y=Probability_Admission2))+geom_point()



#No of students having High GRE(580+)admitted =84
#Probability of students with High GRE getting admitted= 84/127= 66.14 %

#No of students having  Medium GRE(440-580)admitted=39
#Probability of students with Medium GRE getting admitted= 39/127=30.7%

#No of students having  Low GRE(0-440)admitted=4
##Probability of students with Low GRE getting admitted=4/127= 3.14%
```