#Admission Dataset

#load packages

```
library("MLmetrics") #metric for model
library(readxl)#to read excel
library(plyr)#split and put data
library(DescTools) #For Mode
library(dplyr) #For Pipe Operator
library(ggplot2) #For QQ Plot
library(moments) #For Skewness & Kurtosis
library(rstatix) #For Welch ANOVA Test
library(Hmisc) #For rcorr Function & Missing Values Treatment
library(QuantPsyc) #For lm.beta function
library(ggpubr) #For advanced QQ Plots
library(aod) #For wald.test function in logistic regression
library(fmsb)#For NagelkerkeR2 function in logistic regression
library(caret) #For Data manupilation
library(imputeMissings) #For imputing missing Values
library(purrr) #For Missing Values
library(naivebayes) #For naive bayes
library(rpart.plot) #For DT Graph
library(psych) #descriptive stats
library(rpart) #for DT
library(e1071) #for SVM
library(knitr) #to save as pdf
library(gridExtra)#work with grid graphics
library(datasets)#dataset related operations
library(tinytex)# to convert to latex
```

```
#load and attach Admissions dataset

AdmissionData<- read_xlsx("C:\\Users\\sujoydutta\\Desktop\\Data analysis\\R\\Datasets_R\\Lesson 8_Classification\\Admission data.xlsx")
str(AdmissionData)

#converting to dataframe
Add<-data.frame(AdmissionData)
str(Add)

#####################################################################
#converting variables to appropriate types

Add$admit<-as.factor(AdmissionData$admit)
Add$rank<-as.factor(AdmissionData$rank)
Add$gre<-as.numeric(AdmissionData$gre)
Add$gpa<-as.numeric(AdmissionData$gpa)
str(Add)

#####################################################################
# check for missing values
map(Add, ~sum(is.na(.)))


#Checking Skewness

sapply(Add[(2:3)], "skewness")


#boxplot to see outliers
boxplot(Add$gre)
```

```
boxplot(Add$gpa)


#getting the outliers
outA<-boxplot(Add$gre)$out
outB<-boxplot(Add$gpa)$out


#removing the outliers
Addnew<- Add
Addnew<-Addnew[-which(Add$gre %in% outA),]
Addnew<-Addnew[-which(Add$gpa %in% outB),]
str(Addnew)
#####################################################################
#categorizing gre
Addnew<- transform(Addnew,GRE_lvl = ifelse(gre<440,"low gre",ifelse(gre<580,"medium gre","high gre")))
str(Addnew)



#summary stats
summary(Addnew)
describe(Addnew)


#normality of new dataset
skewness(Addnew$gre)
kurtosis(Addnew$gre)
skewness(Addnew$gpa)
kurtosis(Addnew$gpa)



#correlation of gpa and gre
ggplot(Addnew, aes(x =gpa, y = gre)) +
```

```
  geom_line() +
  labs(title = "GPA and GRA relation",
       x = "GPA",
       y = "GRA")
corr.test(Addnew$gpa,Addnew$gre,method = "pearson", use = "complete.obs")

# GRE_lvl distribution
ggplot(Addnew, aes(x = GRE_lvl)) +
  geom_density(fill = "indianred3") +
  labs(title = "Participants by GRE_lvl")

# seeing the race distribution

plotdata <- Addnew %>%
  count(Race) %>%
  arrange(desc(Race)) %>%
  mutate(prop = round(n*100/sum(n), 1),
         lab.ypos = cumsum(prop) - 0.5*prop)

plotdata$label <- paste0(plotdata$Race, "\n",
                         round(plotdata$prop), "%")

ggplot(plotdata,
       aes(x = "",
           y = prop,
           fill =Race)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "red") +
  geom_text(aes(y = lab.ypos, label = label),
            color = "yellow") +
```

```
  coord_polar("y",
              start = 0,
              direction = -1) +
  theme_void() +
  theme(legend.position = "FALSE") +
  labs(title = "Race Distribution")
# seeing GRE levels distribution by Rank
ggplot(Addnew,
       aes(x = rank,
           fill = GRE_lvl)) +
  geom_bar(position = "dodge")

#To plot GRE per Social economic level
plot(Addnew$ses,Addnew$gre, xlab = "Number of prospects", ylab = "duration  per prospect", col =
"blue")

#  Calculate test statistics
add_aov <- aov(Addnew$gre~factor(Addnew$ses))
summary(add_aov)

#seeing which gender gets more admission
ggplot(Addnew, aes(x =  Gender_Male, fill = admit)) +
  geom_bar(position = "fill") +
  theme_classic()+
  labs(x = "Gender",y = "Count",fill = "Admission")


# Seeing if high gpa leads to person taking the admission?
boxplot(Addnew$gpa~factor(Addnew$admit),
        xlab = "Did they take the admission?", ylab = "GPA")
```

```
#  Calculate test statistics
add_aov <- aov(Addnew$gpa~factor(Addnew$admit))
summary(add_aov)

str(Addnew)
#####################################################################
#Data partition
set.seed(123)
Train <- createDataPartition(Addnew$admit, p=0.7, list=FALSE)
training <- Addnew[ Train, ]
testing <- Addnew[ -Train, ]

#####################################################################
#Model1- Logistic Regression
#create the model using train()

Model1 <- train(admit~.,  data=training, method="glm", family="binomial")
summary(Model1)

#exponentiated coefficients
exp(coef(Model1$finalModel))

# prediction for test data
predadmit<-predict(Model1, newdata=testing)

#fitness metrics for validation
confusionMatrix(predadmit, testing$admit, mode = "everything", positive="1")

#predicting probability for new values
newvalues<-data.frame(gre=800,gpa=4,ses=2,GRE_lvl='medium
gre',ses=as.factor(2),Gender_Male=1,Race=1,rank=as.factor(1))
```

```
pred<-predict(Model1, newdata=newvalues,type="prob")
pred



#####################################################################
#Model2- SVM
#create the model using train()
Model2<-svm(admit~.,data=training)
summary(Model2)



# prediction for test data
predadmit<-predict(Model2, newdata=testing)

#fitness metrics for validation
confusionMatrix(predadmit, testing$admit, mode = "everything", positive="1")

#predicting probability for new values
newvalues<-data.frame(gre=900,gpa=3.8,ses=2,GRE_lvl='high
gre',Gender_Male=1,Race=2,rank=as.factor(3))
pred<-predict(Model2, newdata=newvalues,type="prob")
pred

#Model3- Decision Tree
#create the model using train()

Model3 <- train(admit~.,  data=training, method="rpart",parms= list(split="information"))
summary(Model3)
```

```
# prediction for test data

predadmit<-predict(Model3, newdata=testing)


#fitness metrics for validation

confusionMatrix(predadmit, testing$admit, mode = "everything", positive="1")


#predicting probability for new values

newvalues<-data.frame(gre=650,gpa=4,ses=1,GRE_lvl='medium
gre',Gender_Male=1,Race=3,rank=as.factor(4))

pred<-predict(Model2, newdata=newvalues,type="prob")

pred


#converting to pdf


knitr::stitch('Admissiondata.r')



setwd("C:\\Users\\sujoydutta\\Desktop\\Data analysis\\R\\R scripts")


dir()

"Admissiondata.R"

knitr::stitch('Admissiondata.R')
```