COLLEGE ADMISSION

#DESCRIPTION

#Background and Objective:

#Every year thousands of applications are being submitted by international students for admission in colleges of the USA.

#It becomes an iterative task for the Education Department to know the total number of applications received

#and then compare that data with the total number of applications successfully accepted and visas processed.

#Hence to make the entire process easy, the education department in the US analyze the factors that influence the admission of a student into colleges.

#The objective of this exercise is to analyse the same.

#Domain: Education

#Dataset Description:

#Attribute	**Description**
#GRE	Graduate Record Exam Scores
#GPA	Grade Point Average
#Rank	It refers to the prestige of the undergraduate institution. #The variable rank takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.
#Admit	It is a response variable; admit/don't admit is a binary variable where 1 indicates that student is admitted and 0 indicates that student is not admitted.
#SES	SES refers to socioeconomic status: 1 - low, 2 - medium, 3 - high.
#Gender_male	Gender_male (0, 1) = 0 -> Female, 1 -> Male
#Race	Race – 1, 2, and 3 represent Hispanic, Asian, and African-American

########## **SOURCE CODE** ##############

```
rm(list=ls(all= TRUE)) #clearing the environment

library(corrplot)
library(caret)
library(ggplot2)
library(caTools)
library(MASS)
library(kernlab)
library(coefplot)

data_file <- read.csv("College_admission.csv") #reading the dataset
View(data_file)
```

########## **EXPLORATORY DATA ANALYSIS** ##########

```
head(data_file)
tail(data_file)
str(data_file)
class(data_file)
dim(data_file)
summary(data_file)
#from observing the dataset:-
```

#the dependent variable is found to be : ADMIT

#independent variables are : gre, gpa, ses, Gender_Male, race, rank

########## PREDICTIVE ##########

#find the missing value

```
sum(is.na(data_file))
#sum is 0, hence there are no missing values
```

#find outliers

```
#using boxplot to find the outliers in the continuous variables gpa and gre.

boxplot(data_file$gre, main="boxplot of gre", horizontal = T)
boxplot.stats(data_file$gre)$out # gre contains 4 outliers(300,300,220,300)

boxplot(data_file$gpa, main="boxplot of gpa", horizontal = T)
boxplot.stats(data_file$gpa)$out #gpa contain one outlier(2.26)

out <- boxplot.stats(data_file$gre)$out #initializing out variable with outlier values
out_ind <- which(data_file$gre %in% c(out)) #storing the index of outliers
out_ind #outliers are found in the following rows 72,780,305,316

data_file[out_ind,]
data_file <- data_file[-c(72,180,305,316),] #dropping the rows with outliers
dim(data_file) #now we have 396 records
View(data_file)

#performing similar operation for gpa
out2 <- boxplot.stats(data_file$gpa)$out
out_ind2 <- which(data_file$gpa %in% c(out2))
out_ind2
```

```
data_file[out_ind2,]
data_file <- data_file[-c(290),]
dim(data_file)
View(data_file)
```

#Find the structure of the data set and if required, transform the numeric data type to factor and vice-versa.

```
str(data_file) #using str() function to find the structure of the dataset

#converting categorical variables into factors
data_file$admit <- as.factor(data_file$admit)
data_file$ses <- as.factor(data_file$ses)
data_file$Gender_Male <- as.factor(data_file$Gender_Male)
data_file$Race <- as.factor(data_file$Race)
data_file$rank <- as.factor(data_file$rank)
str(data_file)
```

#Find whether the data is normally distributed or not. Use the plot to determine the same.

```
#using histogram to check the normality
hist(data_file$gre)

#using density plot
plot(density(data_file$gre))

hist(data_file$gpa)
plot(density(data_file$gpa))
```

```
#the visualization output shows that the datas are normally distributed.
#however the values of gre and gpa are in different scales,so scaling is necessary in this case.
data_file2 <- data_file
data_file$gre <- scale(data_file$gre, center = T, scale = T)
data_file$gpa <- scale(data_file$gpa, center = T, scale = T)
head(data_file)
```

#Use variable reduction techniques to identify significant variables.

```
#we can find significant variables by building a regression model.
plot(data_file)
#We can build logistic regression model and identify significant variables.
#str(data_file)

###splitting data set ###
set.seed(123)
indices <- sample.split(data_file, SplitRatio = 0.7)
train <- data_file[indices == T,]
test <- data_file[indices == F,]
View(train)
View(test)
dim(train)
dim(test)

### building model ###
rmodel <- glm(admit~ ., data = train, family = "binomial")
summary(rmodel)
newdata1 <- with(data_file, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
newdata1
coefplot(rmodel)
```

```
#After looking at the summary results,the significant variables can be identified as GRE, GPA and RANK

#dropping insignificant variables.

final_file <- data_file[,c(1,2,3,7)]

View(final_file)

class(final_file)

head(final_file)
```

#Calculate the accuracy of the model and run validation techniques.

```
set.seed(123)

ind <- sample(1:nrow(final_file), 0.70 * nrow(final_file))

Train <- final_file[ind, ]

Test <- final_file[-ind,]

test_without_admit <- Test[,-1]

View(test_without_admit)

log_reg <- glm(admit ~ . , data =Train,
               family = 'binomial')


summary(log_reg)

coefplot(log_reg)



#accuracy of the model can be calculated using confusion matrix.

prob_train <- predict(log_reg,test_without_admit, type = "response")

preds_train <- ifelse(prob_train > 0.49,1,0)# use 0.5 or 0.49 to get the best accuracy

comp = table(Test$admit,preds_train)

confusionMatrix(comp,positive = "0")


#the model accuracy is 74% .
```

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

SUPPORT VECTOR MACHINES

```
library(e1071)
model_svm = svm(admit~.,Train,kernel = "linear")
summary(model_svm)
predictn <- predict(model_svm,test_without_admit,type = "Class")
confusionMatrix(predictn,Test$admit)
#Accuracy : 0.7311
```

DECISION TREE

```
library(rpart)
library(rpart.plot)
library(randomForest)

tree_model <- rpart(admit~. , data = Train, method = "class")
class(tree_model)
prp(tree_model)
rpart.plot(tree_model)

pred_class <- predict(tree_model,test_without_admit, type = 'class')
confusionMatrix(pred_class, Test$admit, positive = '0')
#Accuracy : 0.7059
```

RANDOM FOREST

```
memory.limit(size = 56000)
set.seed(71)
rand_f <- randomForest(admit~. , data = Train, ntree =30,mtry=2, na.action = na.omit)
rand_f
tpred <- predict(rand_f,newdata = Test)
confusionMatrix(tpred,Test$admit)
#   Accuracy : 0.7227
```

#Determine the accuracy rates for each kind of model.

#the accuracy rates for each model is as follows:

#1.Logistic Regression : 74.79%

#2.Decision Tree (rpart) : 70.59%

#3.Random Forest : 72.22%

#4.SVM : 73.11%

#Select the most accurate model

#by evaluating the accuracy of different models:

#the most accurate model is logistic regression(74.79%)

####Descriptive:

#Categorize the average of grade point into High, Medium, and Low (with admission probability percentages) and plot it on a point chart.

#the probability for gpa in logistic regression is found to be 0.042(4.2%)

#college_dat <- read.csv("College_admission.csv")

#View(college_dat)

#head(college_dat)

View(data_file)

summary(data_file$gpa)

Im unable to complete the descriptive part, because, i dont remember similar operation being taught.upon raising a ticket and discussion with support team,i was provided with a video which explains this project, but the descriptive part was left behind in the same. kindly consider this and i would like to know the solution for this .