**Education project in R:**

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 |

**Analysis Tasks:**Analyze the historical data and determine the key drivers for admission.

**Predictive:**

* Find the missing values. (if any, perform missing value treatment)
* Find outliers (if any, then perform outlier treatment)
* Find the structure of the data set and if required, transform the numeric data type to factor and vice-versa.
* Find whether the data is normally distributed or not. Use the plot to determine the same.
* Normalize the data if not normally distributed.
* Use variable reduction techniques to identify significant variables.
* Run logistic model to determine the factors that influence the admission process of a student (Drop insignificant variables)
* Calculate the accuracy of the model and run validation techniques.
* Try other modelling techniques like decision tree and SVM and select a champion model
* Determine the accuracy rates for each kind of model
* Select the most accurate model
* Identify other Machine learning or statistical techniques

**Descriptive:**  
Categorize the average of grade point into High, Medium, and Low (with admission probability percentages) and plot it on a point chart.    
Cross grid for admission variables with GRE Categorization is shown below:

|  |  |
| --- | --- |
| **GRE** | **Categorized** |
| 0-440 | Low |
| 440-580 | Medium |
| 580+ | High |

***Business scenario:*** *To analyze the factors that influence the admission of a student into college.*

***Expectation/Goals:*** *Run logit model, drop insignificant variables, accuracy rates of each model, convert variables into factors*

***Code:***

*Logistic regression algorithm: SVM algorithm:*

library(readr)

Project\_1\_Dataset1<-read.csv("Practice datasets/Project 1\_Dataset.csv")

View(Project\_1\_Dataset1)

str(Project\_1\_Dataset1)

Project\_1\_Dataset1$admit<-as.factor(Project\_1\_Dataset1$admit)

Project\_1\_Dataset1$rank<-as.factor(Project\_1\_Dataset1$rank)

Project\_1\_Dataset1$Race<-as.factor(Project\_1\_Dataset1$Race)

Project\_1\_Dataset1$ses<-as.factor(Project\_1\_Dataset1$ses)

Project\_1\_Dataset1$Gender\_Male<-as.factor(Project\_1\_Dataset1$Gender\_Male)

str(Project\_1\_Dataset1)

library(caTools)

split<-sample.split(Project\_1\_Dataset1,SplitRatio=0.8)

split

Training<-subset(Project\_1\_Dataset1,split=="TRUE")

Testing<-subset(Project\_1\_Dataset1,split=="FALSE")

Model<-glm(admit ~.,Training, family="binomial")

summary(Model)

Res<-predict(Model, Training, type="response")

Res

Res<-predict(Model, Testing, type="response")

Confusionmatrix<-(table(Actual\_Value=Training$admit,Predicted\_Value = Res > 0.5))

Confusionmatrix

(Confusionmatrix[[1,1]]+Confusionmatrix[[2,2]])/sum(Confusionmatrix)

71.2% accuracy

gpa,Race-significant

ses,Gender\_Male-insignificant

Model\_1<-glm(admit ~.-(“ses”+”Gender\_Male”),Training, family="binomial")

summary(Model)

#to find out the max threshold:

Res<-predict(Model, Training, type="response")

install.packages("ROCR")

library(ROCR)

ROCRPred<-prediction(Res,Training$admit)

ROCRPref<-performance(ROCRPred,"tpr","fpr")

plot(ROCRPref,colorize=TRUE,print.cutoffs.at=seq(0.1,by=0.1))

#max threshold is found at 0.4

Confusionmatrix<-(table(Actual\_Value=Training$admit,Predicted\_Value = Res > 0.4))

Confusionmatrix

(Confusionmatrix[[1,1]]+Confusionmatrix[[2,2]])/sum(Confusionmatrix)

71.9%accuracy

#Categorize the gpa:

library(readr)

Project\_1\_Dataset1<-read.csv("Practice datasets/Project 1\_Dataset.csv")

View(Project\_1\_Dataset1)

str(Project\_1\_Dataset1)

library(dplyr)

options(max.print=1000000)

options

res<-Project\_1\_Dataset1 %>% mutate(category=cut(gre,breaks = c(-Inf,440,580,Inf),labels = c("Low","Medium","High")))

res

#Categorize the gre:

library(readr)

Project\_1\_Dataset1<-read.csv("Practice datasets/Project 1\_Dataset.csv")

View(Project\_1\_Dataset1)

str(Project\_1\_Dataset1)

library(dplyr)

options(max.print=1000000)

options

res<-Project\_1\_Dataset1 %>% mutate(category=cut(gpa,breaks = c(-Inf,3.22,4,Inf),labels = c("Low","Medium","High")))

res

install.packages("caret")

library(caret)

library(readr)

Project\_1\_Dataset1<-read.csv("Practice datasets/Project 1\_Dataset.csv")

View(Project\_1\_Dataset1)

set.seed(3033)

Sample\_split<-floor(0.7\*nrow(Project\_1\_Dataset1))

training<-createDataPartition(y=Project\_1\_Dataset1$admit,p=0.7,list=FALSE)

Train<-Project\_1\_Dataset1[training,]

Test<-Project\_1\_Dataset1[-training,]

dim(Train)

dim(Test)

Train[["admit"]]=factor(Train[["admit"]])

trctrl<-trainControl(method="repeatedcv",number=10,repeats=3)

svm\_linear<-train(admit ~.,data=Train,method="svmLinear",trControl=trctrl,preProcess=c("center","scale"),tuneLength=10)

svm\_linear

install.packages("e1071")

test\_pred<-predict(svm\_linear,newdata=Test)

test\_pred

confusionMatrix(table(test\_pred,Test$admit))

grid<-expand.grid(C=c(0,0.01,0.05,0.1,0.25,0.5,0.75,1,1.25,1.5,1.75,2.5))

svm\_linear\_grid<-train(admit ~.,data=Train,method="svmLinear",trControl=trctrl,preProcess=c("center","scale"),tuneGrid=grid,tuneLength=10)

svm\_linear\_grid

plot(svm\_linear\_grid)

test\_pred<-predict(svm\_linear\_grid,newdata=Test)

test\_pred

confusionMatrix(table(test\_pred,Test$admit))

69.17% accuracy

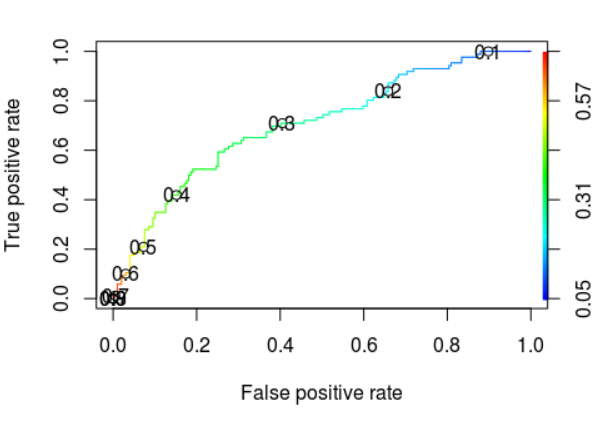
***Output Screenshot:***

***Analysis:***

1. From logit model algorithm the accuracy is tested as **71.2%** intitially after optimizing the model the accuracy increased to **71.9%**
2. From SVM algorithm the accuracy is tested as **69.17%** even after optimizing the model could not see any difference in the accuracy
3. I feel Logit model is more accurate and easy to perform compared to SVM however each has its own advantages and disadvantages.

gpa,Race-Significant variables

ses,Gender\_Male-Insignificant Variables

**

*\*\*The above boxes contain the whole code. Please drag the boxes to view it.\*\**