European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

PREDICTING STUDENT UNIVERSITY ADMISSION USING LOGISTIC REGRESSION

Sharan Kumar Paratala Rajagopal
Senior Manager, Capgemini America Inc., Dallas, USA
Email: prsharankumar@gmail.com

ABSTRACT*: The primary purpose is to discuss the prediction of student admission to university based on numerous factors and using logistic regression. Many prospective students apply for Master's programs. The admission decision depends on criteria within the particular college or degree program. The independent variables in this study will be measured statistically to predict graduate school admission. Exploration and data analysis, if successful, would allow predictive models to allow better prioritization of the applicants screening process to Master's degree programme which in turn provides the admission to the right candidates.*

KEYWORDS*:* logistic regression, predictive analysis, college admission, data analytics

INTRODUCTION

Student admission for the Master's degree program consists of different criteria/scores which is taken into consideration before admitting the student to the degree program. This process is elaborative and requires lot of thought processing and analysis by the selection committee before choosing the right applicants to the Master's degree program.

The purpose of this analysis is to demonstrate the top contributing scores which helps the student to get the admission into the Master's degree program. What factors contributes to successful admission to a Master's degree program?

The analysis might seem straight forward but caution has to be exercised to consider the scores like GRE, TOEFL, university rating, SOP, LOR and CGPA and any outliers should not impact the decision making process.

Data Collection

The subject analysis will require the collection and generation of data from UCLA Graduate Dataset. The existing data set will be used for analysis and predicting the factors which will influence for the admission process. [1]

This dataset is created for prediction of Graduate Admissions from an Indian perspective. Many prospective students apply for UCLA Master's programs. The admission decision depends on criteria within the particular college or degree program. The independent variables in this study will be measured statistically to predict graduate school admission. Exploration and data analysis, if successful, would allow predictive models to allow better prioritization of the applicants screening process to Master's degree program which in turn provides the admission to the right candidates.

European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

Data Extraction and Preparation

UCLA data set will be examined for predictor variables which contribute to the college admission process. Data cleansing will be performed to eliminate irrelevant duplicates and outliers. The dataset consists of the below variables. "Table 1" provides the details of the variables if its predictor or response variables.

- GRE Scores (out of 340)
- TOEFL Scores (out of 120)
- University Rating (out of 5)
- Statement of Purpose and
- Letter of Recommendation Strength (out of 5)
- Undergraduate GPA (out of 10)
- Research Experience (either 0 or 1)
- Chance of Admit (ranging from 0 to 1)

Variable Name	Type	Predictor/Response
Serial No.	Continuous	Predictor
GRE Score	Continuous	Predictor
TOEFL Score	Continuous	Predictor
University Rating	Continuous	Predictor
SOP	Continuous	Predictor
LOR	Continuous	Predictor
CGPA	Continuous	Predictor
Research	Binary	Predictor
Chance of Admit	Binary/Categorical	Response

Table 1. Variable and type definition

Tools to Import Data

Raw data [2] will be extracted into .csv format from the reference [1] and RStudio import wizard will be used to import data set from .csv file and will be performing the Logistic regression on the data set.

Logistic regression is used to model the relationship between a binary response variable and a set of predictor variables. It's used to estimate the probability of the response according to the various continuous and categorical predictors. The estimated probabilities can then be used to classify an unknown response into one of the two outcome levels, given a set of predictors.

First will be looking for associations between your predictors, such as number of GRE, TOFEL, SOP, LOA and the binary response Chance of Admit, to see which variables should be considered for model inclusion. Then will use logistic regression to determine which students will have high probability of getting admission to Master's program.

European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

After importing there are 400 observations and 9 variables as shown in "Figure1".

	Serial.No.	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	Chance.of.Admit
1	1	337	118	4	4.5	4.5	9.65	1	0.92
2	2	324	107	4	4.0	4.5	8.87	1	0.76
3	3	316	104	3	3.0	3.5	8.00	1	0.72
4	4	322	110	3	3.5	2.5	8.67	1	0.80
5	5	314	103	2	2.0	3.0	8.21	0	0.65
6	6	330	115	5	4.5	3.0	9.34	1	0.90
7	7	321	109	3	3.0	4.0	8.20	1	0.75
8	8	308	101	2	3.0	4.0	7.90	0	0.68
9	9	302	102	1	2.0	1.5	8.00	0	0.50
10	10	323	108	3	3.5	3.0	8.60	0	0.45
11	11	325	106	3	3.5	4.0	8.40	1	0.52
12	12	327	111	4	4.0	4.5	9.00	1	0.84

Figure 1 Raw data observations

Chance of Admit is the binary/categorical variable which is the response variable and helps in prediction of the admission to Master's degree program.

Load dataset:

```
> df <- read.csv("~/MSDA/C772/project/Admission_Predict.csv")
> dff <- df
```

Checking duplicate rows

```
> length(unique(df$Serial.No.)) == nrow(df)
[1] TRUE
```

Check for missing values

```
> sum(is.na(df))
[1] 0
```

European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

Summary of data

```
> summary(df)
   Serial.No.       GRE.Score      TOEFL.Score     University.Rating      SOP
 Min.   :  1.0   Min.   :290.0   Min.   : 92.0   Min.   :1.000      Min.   :1.0
 1st Qu.:100.8   1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000      1st Qu.:2.5
 Median :200.5   Median :317.0   Median :107.0   Median :3.000      Median :3.5
 Mean   :200.5   Mean   :316.8   Mean   :107.4   Mean   :3.087      Mean   :3.4
 3rd Qu.:300.2   3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000      3rd Qu.:4.0
 Max.   :400.0   Max.   :340.0   Max.   :120.0   Max.   :5.000      Max.   :5.0
      LOR             CGPA          Research      Chance.of.Admit
 Min.   :1.000   Min.   :6.800   Min.   :0.0000   Min.   :0.3400
 1st Qu.:3.000   1st Qu.:8.170   1st Qu.:0.0000   1st Qu.:0.6400
 Median :3.500   Median :8.610   Median :1.0000   Median :0.7300
 Mean   :3.453   Mean   :8.599   Mean   :0.5475   Mean   :0.7244
 3rd Qu.:4.000   3rd Qu.:9.062   3rd Qu.:1.0000   3rd Qu.:0.8300
 Max.   :5.000   Max.   :9.920   Max.   :1.0000   Max.   :0.9700
```

Data Cleaning

Serial # does not affect the chance of admission. Hence making the value to Null.

```
df$Serial.No. = NULL
```

Analysis of Variables

Analyzing the variable GRE.Score

```
> summary(df$GRE.Score)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  290.0   308.0   317.0   316.8   325.0   340.0
> quantile(df$GRE.Score, seq(0,1,0.01))
    0%     1%     2%     3%     4%     5%     6%     7%     8%     9%    10%    11%    12%
290.00 294.00 295.00 296.00 297.00 298.00 298.00 299.00 299.00 300.00 300.00 301.00 301.00
   13%    14%    15%    16%    17%    18%    19%    20%    21%    22%    23%    24%    25%
302.00 303.00 303.85 304.00 304.83 305.00 305.81 306.00 307.00 307.00 308.00 308.00 308.00
   26%    27%    28%    29%    30%    31%    32%    33%    34%    35%    36%    37%    38%
309.00 310.00 310.00 311.00 311.00 311.00 312.00 312.00 312.00 312.00 312.00 313.00 313.00
   39%    40%    41%    42%    43%    44%    45%    46%    47%    48%    49%    50%    51%
313.00 314.00 314.00 314.00 315.00 315.00 315.00 316.00 316.00 316.00 317.00 317.00 317.00
   52%    53%    54%    55%    56%    57%    58%    59%    60%    61%    62%    63%    64%
318.00 318.00 318.46 319.00 319.00 320.00 320.00 320.00 321.00 321.00 321.00 322.00 322.00
   65%    66%    67%    68%    69%    70%    71%    72%    73%    74%    75%    76%    77%
322.00 322.34 323.00 323.00 324.00 324.00 324.00 324.00 324.00 325.00 325.00 325.00 326.00
   78%    79%    80%    81%    82%    83%    84%    85%    86%    87%    88%    89%    90%
326.00 326.00 327.00 327.00 327.00 328.00 329.00 329.00 329.14 330.00 331.00 331.00 332.00
   91%    92%    93%    94%    95%    96%    97%    98%    99%   100%
332.09 333.08 334.00 335.00 336.00 336.04 338.00 339.02 340.00 340.00
> q1 <- quantile(df$GRE.Score, c(0.25))
> q3 <- quantile(df$GRE.Score, c(0.75))
> IQR <- q3 - q1
> upper_range <- q3 + 1.5*IQR
> lower_range <- q1 - 1.5*IQR
> nrow(df[df$GRE.Score > upper_range,])
[1] 0
> nrow(df[df$GRE.Score < lower_range,])
[1] 0
```

Vol.8, No.3, pp.46-56, June 2020

Published by ***ECRTD- UK***

Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

Analyzing the variable TOEFL.score

```
> summary(df$TOEFL.Score)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   92.0   103.0   107.0   107.4   112.0   120.0
> quantile(df$TOEFL.Score, seq(0,1,0.01))
    0%     1%     2%     3%     4%     5%     6%     7%     8%     9%    10%    11%    12%
 92.00  94.99  96.00  97.00  97.96  98.00  98.00  99.00  99.00  99.00  99.00 100.00 100.00
   13%    14%    15%    16%    17%    18%    19%    20%    21%    22%    23%    24%    25%
100.00 100.00 100.00 101.00 101.00 101.00 102.00 102.00 102.00 102.00 103.00 103.00 103.00
   26%    27%    28%    29%    30%    31%    32%    33%    34%    35%    36%    37%    38%
103.00 104.00 104.00 104.00 104.00 104.00 104.00 105.00 105.00 105.00 105.00 105.00 105.00
   39%    40%    41%    42%    43%    44%    45%    46%    47%    48%    49%    50%    51%
105.00 106.00 106.00 106.00 106.00 106.00 106.00 107.00 107.00 107.00 107.00 107.00 107.00
   52%    53%    54%    55%    56%    57%    58%    59%    60%    61%    62%    63%    64%
107.00 108.00 108.00 108.00 108.00 109.00 109.00 109.00 109.00 110.00 110.00 110.00 110.00
   65%    66%    67%    68%    69%    70%    71%    72%    73%    74%    75%    76%    77%
110.00 110.00 110.00 110.00 110.00 110.30 111.00 111.00 111.00 111.00 112.00 112.00 112.00
   78%    79%    80%    81%    82%    83%    84%    85%    86%    87%    88%    89%    90%
112.00 112.00 113.00 113.00 113.00 113.17 114.00 114.00 114.00 115.00 115.00 115.11 116.00
   91%    92%    93%    94%    95%    96%    97%    98%    99%   100%
116.00 117.00 117.00 118.00 118.00 119.00 119.00 119.02 120.00 120.00
> q1 <- quantile(df$TOEFL.Score, c(0.25))
> q3 <- quantile(df$TOEFL.Score, c(0.75))
> IQR <- q3 - q1
> upper_range <- q3 + 1.5*IQR
> lower_range <- q1 - 1.5*IQR
> nrow(df[df$TOEFL.Score > upper_range,])
[1] 0
> nrow(df[df$TOEFL.Score < lower_range,])
[1] 0
```

Analyzing the variable CGPA

```
> summary(df$CGPA)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.800   8.170   8.610   8.599   9.062   9.920
> quantile(df$CGPA, seq(0,1,0.01))
    0%     1%     2%     3%     4%     5%     6%     7%     8%     9%    10%    11%    12%
6.8000 7.2998 7.3992 7.4600 7.5384 7.6400 7.6500 7.6600 7.7000 7.8364 7.8600 7.8800 7.8900
   13%    14%    15%    16%    17%    18%    19%    20%    21%    22%    23%    24%    25%
7.9000 7.9686 8.0000 8.0000 8.0100 8.0282 8.0400 8.0700 8.1000 8.1200 8.1300 8.1576 8.1700
   26%    27%    28%    29%    30%    31%    32%    33%    34%    35%    36%    37%    38%
8.2000 8.2073 8.2200 8.2400 8.2600 8.2700 8.2936 8.3134 8.3300 8.3400 8.3600 8.4000 8.4200
   39%    40%    41%    42%    43%    44%    45%    46%    47%    48%    49%    50%    51%
8.4300 8.4460 8.4500 8.4600 8.4800 8.5000 8.5110 8.5400 8.5453 8.5600 8.5651 8.6100 8.6400
   52%    53%    54%    55%    56%    57%    58%    59%    60%    61%    62%    63%    64%
8.6448 8.6500 8.6600 8.6700 8.6800 8.6943 8.7242 8.7441 8.7600 8.7600 8.7700 8.7837 8.7936
   65%    66%    67%    68%    69%    70%    71%    72%    73%    74%    75%    76%    77%
8.8000 8.8434 8.8733 8.9000 8.9600 8.9700 9.0000 9.0100 9.0254 9.0400 9.0625 9.0848 9.1000
   78%    79%    80%    81%    82%    83%    84%    85%    86%    87%    88%    89%    90%
9.1100 9.1200 9.1300 9.1419 9.1600 9.1800 9.2032 9.2300 9.2400 9.2800 9.3224 9.3600 9.3820
   91%    92%    93%    94%    95%    96%    97%    98%    99%   100%
9.4309 9.4500 9.4707 9.5306 9.6010 9.6600 9.7000 9.7604 9.8002 9.9200
> q1 <- quantile(df$CGPA, c(0.25))
> q3 <- quantile(df$CGPA, c(0.75))
> IQR <- q3 - q1
> upper_range <- q3 + 1.5*IQR
> lower_range <- q1 - 1.5*IQR
> nrow(df[df$CGPA > upper_range,])
[1] 0
> nrow(df[df$CGPA < lower_range,])
[1] 1
> # 1 outliers in upper range of 'CGPA'
> # Treating outliers
> df$CGPA[which(df$CGPA < lower_range)] <- lower_range
```

Analyzing variable university. Rating

```
> summary(factor(df$University.Rating))
  1   2   3   4   5
 26 107 133  74  60
> df$University.Rating <- as.factor(df$University.Rating)
```

Analyzing variable SOP

```
> summary(factor(df$SOP))
  1 1.5   2 2.5   3 3.5   4 4.5   5
  6  20  33  47  64  70  70  53  37
> df$SOP <- as.factor(df$SOP)
```

Analyzing variable LOR

```
> summary(factor(df$LOR))
  1 1.5   2 2.5   3 3.5   4 4.5   5
  1   7  38  39  85  73  77  45  35
> df$LOR <- as.factor(df$LOR)
```

Analyzing variable Research

```
> summary(factor(df$Research))
  0   1
181 219
> df$Research <- as.factor(df$Research)
```

Create a new variable

Classify data with greater than 0.72 because of 0.5 gives un-leveled data division

```
> df$Research <- as.factor(df$Research)
> table(df$Chance.of.Admit > 0.5) # False = 35, True = 365

FALSE  TRUE
   35   365
> df$get_admission = as.factor(ifelse(df$Chance.of.Admit > 0.72,1,0)) #False = 196, True = 204
> table(df$Chance.of.Admit > 0.72)

FALSE  TRUE
  196   204
```

Correlation of Numeric variables with chance of admit

corr <- cor(df_Numeric_Variable)

corrplot(corr,method = "number",type = "full")

```
> corr <- cor(df_Numeric_Variable)
> corrplot(corr,method = "number",type = "full")
```

“Figure 2” provides the details of exam scores being highly correlated.

European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

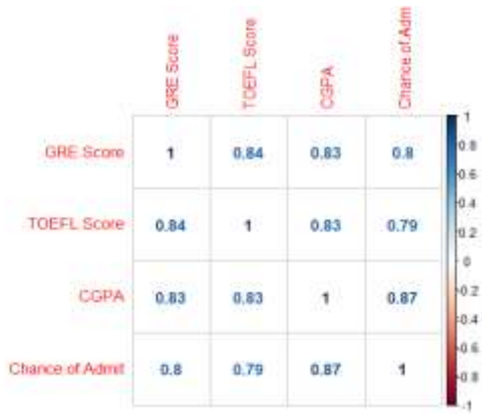


Figure 2 Correlation Plot

From the boxplots it has clearly observed that chance of admission is high when somebody belongs to high ranking university. Although some students are from average rating university, still they have a chance to get admitted.

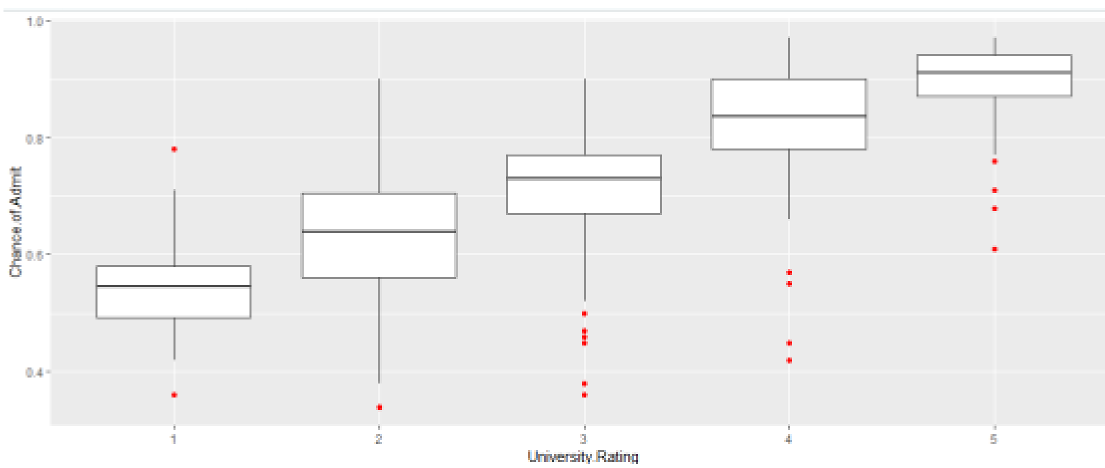


Figure 3 University rating box plot

Split Data Set

Split the data set into training and test data set using below commands

> set.seed(1000)

> indx= sample(1:nrow(df_3), 0.7*nrow(df_3))

> train = df_3[indx,]

> test = df_3[-indx,]

European Journal of Computer Science and Information Technology

Vol.8, No.3, pp.46-56, June 2020

Published by ***ECRTD- UK***

Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

```
> model1.7 <- glm( formula = get_admission ~ GRE.Score + CGPA + Research + LOR5+LOR4.5+Universi
ty.Rating2 + SOP2 + SOP4 + LOR4,family = "binomial", data = train)
> summary(model1.7)#AIC =  168.59, Null deviance =  387.65

Call:
glm(formula = get_admission ~ GRE.Score + CGPA + Research + LOR5 +
    LOR4.5 + University.Rating2 + SOP2 + SOP4 + LOR4, family = "binomial",
    data = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.46185  -0.30360   0.03242   0.31352   2.66356

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -54.17718   10.31117  -5.254 1.49e-07 ***
GRE.Score            0.08588    0.03817   2.250  0.02443 *
CGPA                 3.08356    0.77360   3.986 6.72e-05 ***
Research1            0.92862    0.47763   1.944  0.05187 .
LOR5                 2.63628    1.31003   2.012  0.04418 *
LOR4.5               0.96288    0.76039   1.266  0.20541
University.Rating2  -1.65128    0.62717  -2.633  0.00847 **
SOP2                -0.21820    0.91240  -0.239  0.81099
SOP4                 0.25148    0.53825   0.467  0.64034
LOR4                 0.48757    0.51924   0.939  0.34773
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 387.65  on 279  degrees of freedom
Residual deviance: 148.59  on 270  degrees of freedom
AIC: 168.59

Number of Fisher Scoring iterations: 6
```

Figure 4 Training the model 1.7

“Figure 5” shows less AIC as compared to “Figure 4”

```
> model1.8 <- glm( formula = get_admission ~ GRE.Score + CGPA + Research + LOR5+SOP4.5+Universi
ty.Rating2 + SOP2 + SOP4 + LOR4,family = "binomial", data = train)
> summary(model1.8)#AIC =  168.34, Null deviance =  387.65

Call:
glm(formula = get_admission ~ GRE.Score + CGPA + Research + LOR5 +
    SOP4.5 + University.Rating2 + SOP2 + SOP4 + LOR4, family = "binomial",
    data = train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.38930  -0.27730   0.03349   0.32947   2.69703

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -50.52473   10.30378  -4.904 9.41e-07 ***
GRE.Score            0.07423    0.03827   1.940  0.05243 .
CGPA                 3.08347    0.77192   3.995 6.48e-05 ***
Research1            1.00991    0.48384   2.087  0.03686 *
LOR5                 2.22355    1.30802   1.700  0.08914 .
SOP4.5               1.06916    0.81218   1.316  0.18803
University.Rating2  -1.78115    0.64232  -2.773  0.00555 **
SOP2                -0.12155    0.90236  -0.135  0.89285
SOP4                 0.55199    0.52805   1.045  0.29587
LOR4                 0.27621    0.51620   0.535  0.59259
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 387.65  on 279  degrees of freedom
Residual deviance: 148.34  on 270  degrees of freedom
AIC: 168.34

Number of Fisher Scoring iterations: 6
```

Figure 5 Training the model 1.8

Vol.8, No.3, pp.46-56, June 2020

Published by ***ECRTD- UK***

Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

Confusion Matrix

Create Confusion matrix for Model1.8

predictTrain = predict(model1.8, type="response")s

summary(predictTrain)

table(train$get_admission, predictTrain > 0.5)# Accuracy = 87.86%, FP = 11%, FN = 13%

table(train$get_admission, predictTrain > 0.4)# Accuracy = 87.86%, FP = 14%, FN = 9%

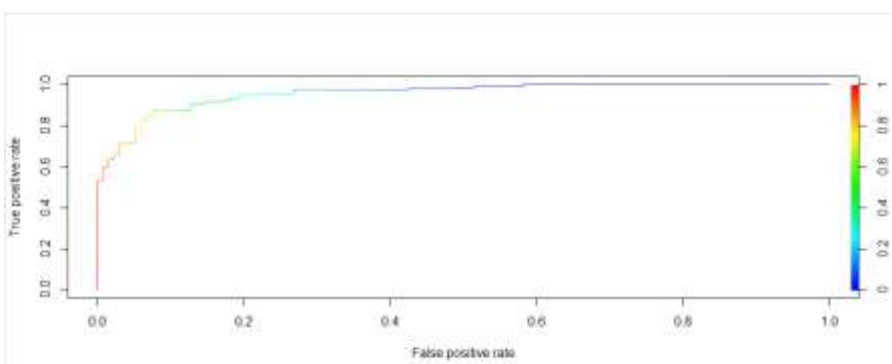
Finalize threshold as 0.5 due to less proportion between False Positive and False Negative

```
> #Model1.8 has less AIC as compared to all other models.
> #Create Confusion matrix for Model1.5
> predictTrain = predict(model1.8, type="response")
> summary(predictTrain)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0003754 0.0564615 0.6053035 0.5214286 0.9612042 0.9997938
> table(train$get_admission, predictTrain > 0.5)# Accuracy = 87.86%, FP = 11%, FN = 13%

    FALSE TRUE
  0   118   16
  1    18  128
> table(train$get_admission, predictTrain > 0.4)# Accuracy = 87.86%, FP = 14%, FN = 9%

    FALSE TRUE
  0   111   23
  1    12  134
> #Finalize threshold as 0.5 due to less proportion between False Positive and False Negative
```

```
> #Build ROC curve for train Set
> pred1 <- prediction(predictTrain,train$get_admission)
> roc.perf = performance(pred1, measure = "tpr", x.measure = "fpr")
> plot(roc.perf,colorize=TRUE)
```




```
> #Test the Model
> predictTest = predict(model1.10, type = "response", newdata = test)
> table(test$get_admission,predictTest >= 0.5)# Accuracy = 87.5%, FP = 7%, FN = 15%

   FALSE TRUE
0     58    4
1     11   47
```

```
> #Build ROC curve for test Set
> pred2 <- prediction(predictTest,test$get_admission)
> roc.perf2 = performance(pred2, measure = "tpr", x.measure = "fpr")
> plot(roc.perf2,colorize=TRUE)
```

European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

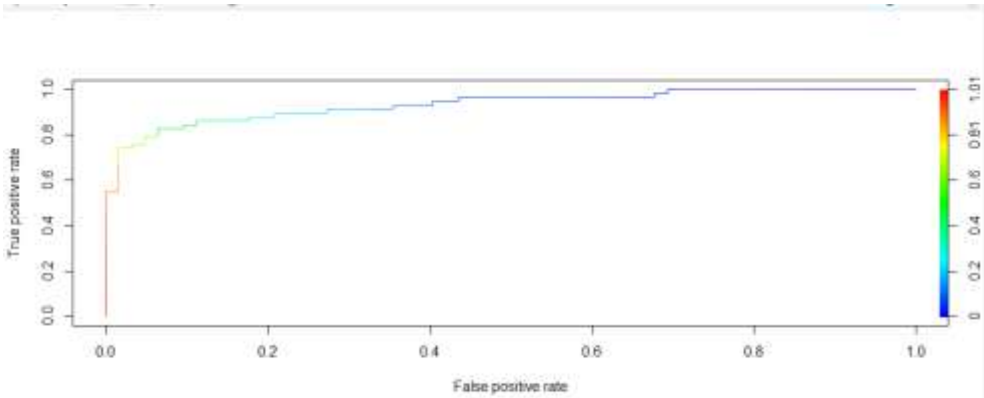


Figure 6 ROC Curve

FINDINGS

The model built is 87.5% accurate to predict admission status of a student. Logistic regression has been used to predict the model.

CONCLUSION

The subject of this examination was to determine if the below variables contribute to the admission of student to Master's degree program.

GRE Score
TOEFL Score
University Rating
SOP
LOR
CGPA

The results of this examination appear to indicate that it greatly contributes to the response variable 'Chance of Admit'. Higher the GRE, TOEFL score then higher the admit chances. The model predicts 87.5% accuracy and can be used for predicting the admit chances based on the above factors. This model will be helpful for the universities to predict the admission and ease their process of selection and timelines.

As part of the hypothesis, the model proved that admission to Master's degree program is dependent on GRE, TOEFL and other scores.

This model would likely be greatly improved by the gathering of additional data of students from different universities which has similar selection criteria to choose the candidates for Master's program.

European Journal of Computer Science and Information Technology
Vol.8, No.3, pp.46-56, June 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

REFERENCES

[1] Mohan S Acharya, Dec 30, 2018, "A Comparison of Regression Models for Prediction of Graduate Admissions, IEEE International Conference on Computational Intelligence in Data Science 2019".

[2] Raw data retrieved from https://www.kaggle.com/mohansacharya/graduate-admissions.