Capstone Project - Graduate Admission

Yu Shing Lui

22 June 2020

1.Introduction

The project performs data analysis and develops a machine learning algorithm to help students in shortlisting universities with their profiles. The predicted output let them have an idea about their opportunity to enter for a particular university.

The dataset contains several parameters which are considered important during the application for Masters Programs. The parameters are including: 1. GRE Scores (out of 340), 2. TOEFL Scores (out of 120), 3. University Rating (out of 5), 4. Statement of Purpose and Letter of Recommendation Strength (out of 5), 5. Undergraduate GPA (out of 10), 6. Research Experience (either 0 or 1), 7. Chance of Admit (ranging from 0 to 1).

This dataset is inspired by the UCLA Graduate Dataset, which are the test scores and GPA are in the older format. Also, it is owned by Mohan S Acharya.

1.Dataset and Package

1.1 Load packages

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
```

1.2 Load dataset

```
library(tidyverse)
library(dplyr)
url <- "https://github.com/yushinglui/graduate_admission/blob/master/datasets_admission.csv?raw=true"
admission <- read.csv(url)
```

2.Data exploration

2.1 General properties of the dataset.

```
head(admission)
```

```
##   Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
## 1          1       337         118                 4 4.5 4.5 9.65        1
## 2          2       324         107                 4 4.0 4.5 8.87        1
## 3          3       316         104                 3 3.0 3.5 8.00        1
## 4          4       322         110                 3 3.5 2.5 8.67        1
## 5          5       314         103                 2 2.0 3.0 8.21        0
## 6          6       330         115                 5 4.5 3.0 9.34        1
##   Chance.of.Admit
## 1            0.92
## 2            0.76
## 3            0.72
## 4            0.80
## 5            0.65
## 6            0.90
```

```
summary(admission)
```

```
##    Serial.No.      GRE.Score      TOEFL.Score    University.Rating
##  Min.   :  1.0   Min.   :290.0   Min.   : 92.0   Min.   :1.000
##  1st Qu.:125.8   1st Qu.:308.0   1st Qu.:103.0   1st Qu.:2.000
##  Median :250.5   Median :317.0   Median :107.0   Median :3.000
##  Mean   :250.5   Mean   :316.5   Mean   :107.2   Mean   :3.114
##  3rd Qu.:375.2   3rd Qu.:325.0   3rd Qu.:112.0   3rd Qu.:4.000
##  Max.   :500.0   Max.   :340.0   Max.   :120.0   Max.   :5.000
##       SOP             LOR             CGPA          Research
##  Min.   :1.000   Min.   :1.000   Min.   :6.800   Min.   :0.00
##  1st Qu.:2.500   1st Qu.:3.000   1st Qu.:8.127   1st Qu.:0.00
##  Median :3.500   Median :3.500   Median :8.560   Median :1.00
##  Mean   :3.374   Mean   :3.484   Mean   :8.576   Mean   :0.56
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:9.040   3rd Qu.:1.00
##  Max.   :5.000   Max.   :5.000   Max.   :9.920   Max.   :1.00
##  Chance.of.Admit
##  Min.   :0.3400
##  1st Qu.:0.6300
##  Median :0.7200
##  Mean   :0.7217
##  3rd Qu.:0.8200
##  Max.   :0.9700
```

2.2 In the dataset, there are 500 rows and 9 columns.

```
dim(admission)
```

```
## [1] 500   9
```

2.3 There are no NA in the dataset.

```
str(admission)
```

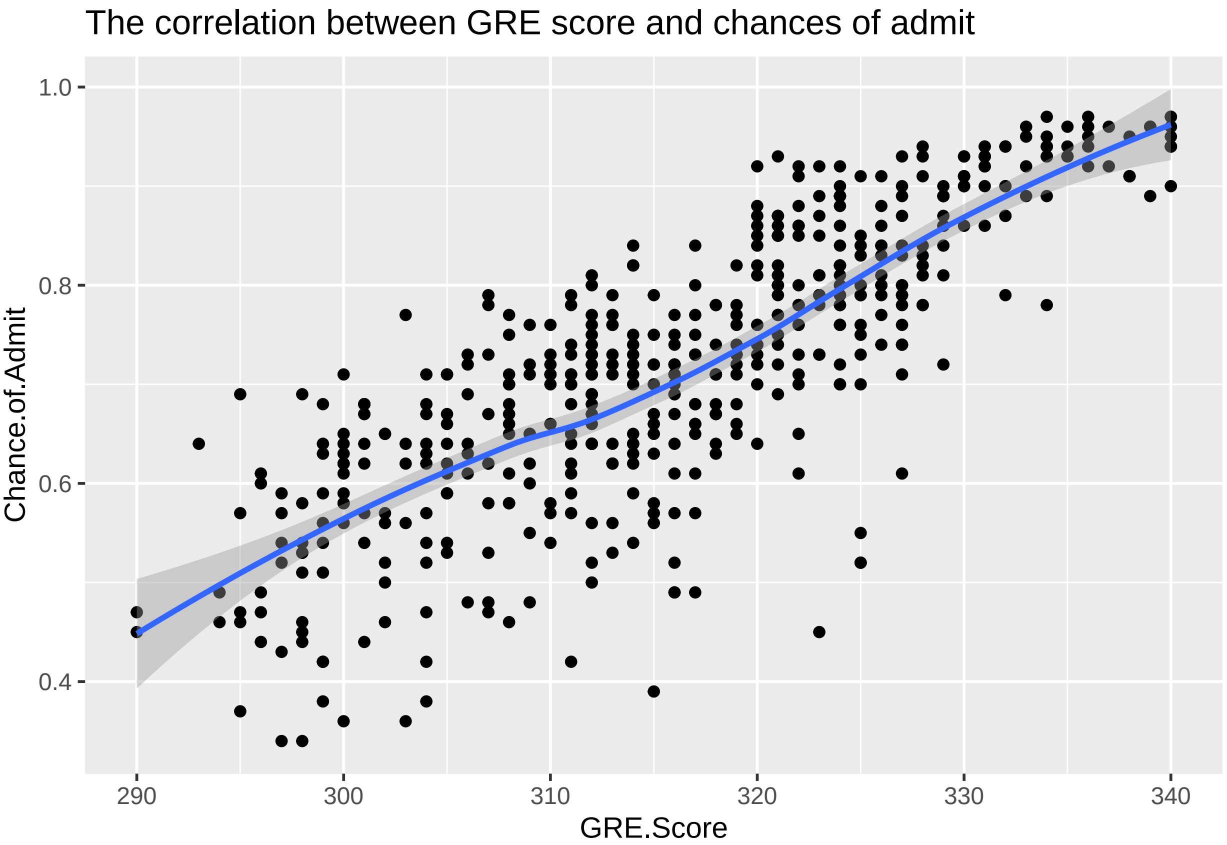
```
## 'data.frame':    500 obs. of  9 variables:
##  $ Serial.No.       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ GRE.Score        : int  337 324 316 322 314 330 321 308 302 323 ...
##  $ TOEFL.Score      : int  118 107 104 110 103 115 109 101 102 108 ...
##  $ University.Rating: int  4 4 3 3 2 5 3 2 1 3 ...
##  $ SOP              : num  4.5 4 3 3.5 2 4.5 3 3 2 3.5 ...
##  $ LOR              : num  4.5 4.5 3.5 2.5 3 3 4 4 1.5 3 ...
##  $ CGPA             : num  9.65 8.87 8 8.67 8.21 9.34 8.2 7.9 8 8.6 ...
##  $ Research         : int  1 1 1 1 0 1 1 0 0 0 ...
##  $ Chance.of.Admit  : num  0.92 0.76 0.72 0.8 0.65 0.9 0.75 0.68 0.5 0.45 ...
```

```
sum(is.na(admission))
```

```
## [1] 0
```

2.4 The diagram shows the relation between GRE score and chance of admit.

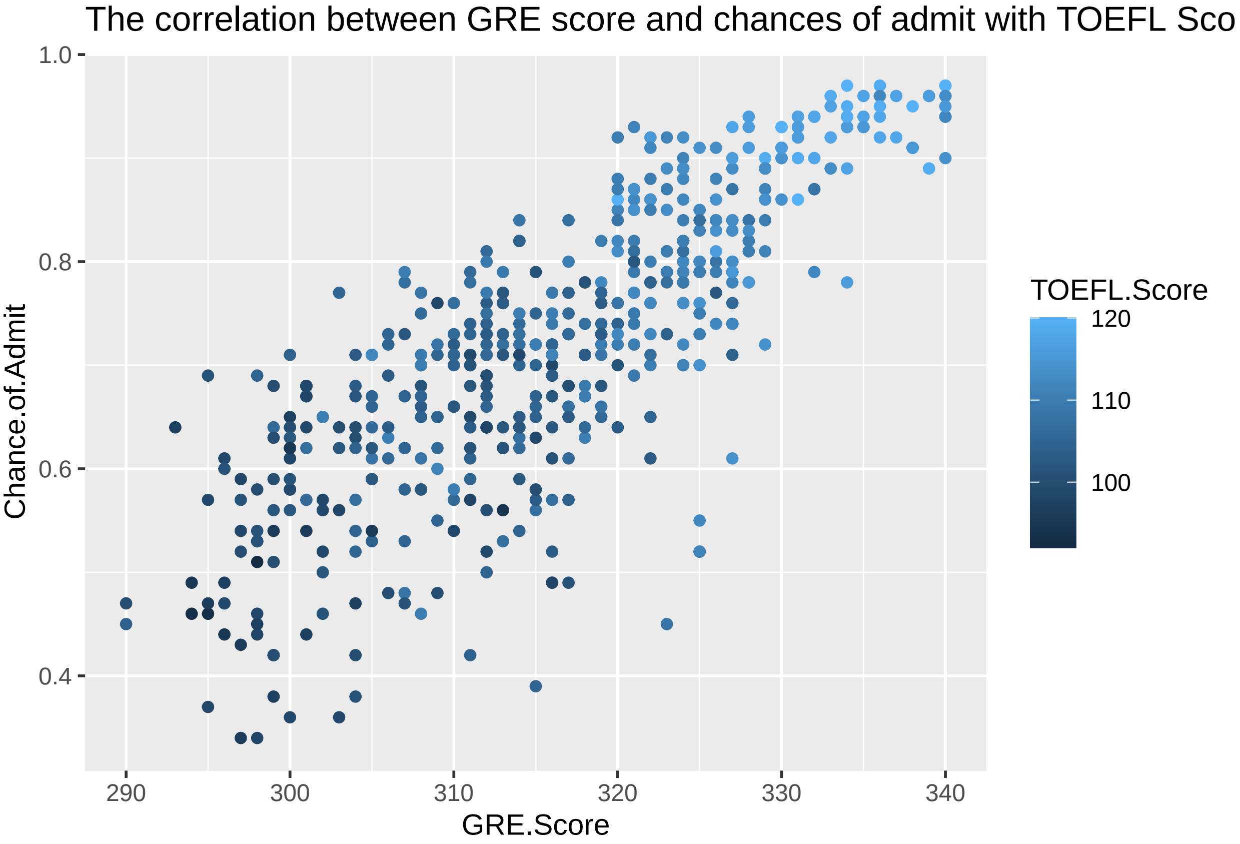
```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit))+geom_point()+geom_smooth()+ggtitle(
  "The correlation between GRE score and chances of admit")
```



The diagram let us know about the GRE score will affect the chance of admission. However, the diagram is not strong enough to show the relationship between them. Now we have to plot some diagrams with the predictors, which is like TOFEL score, University rating, SOP, LOR, and CGPA.

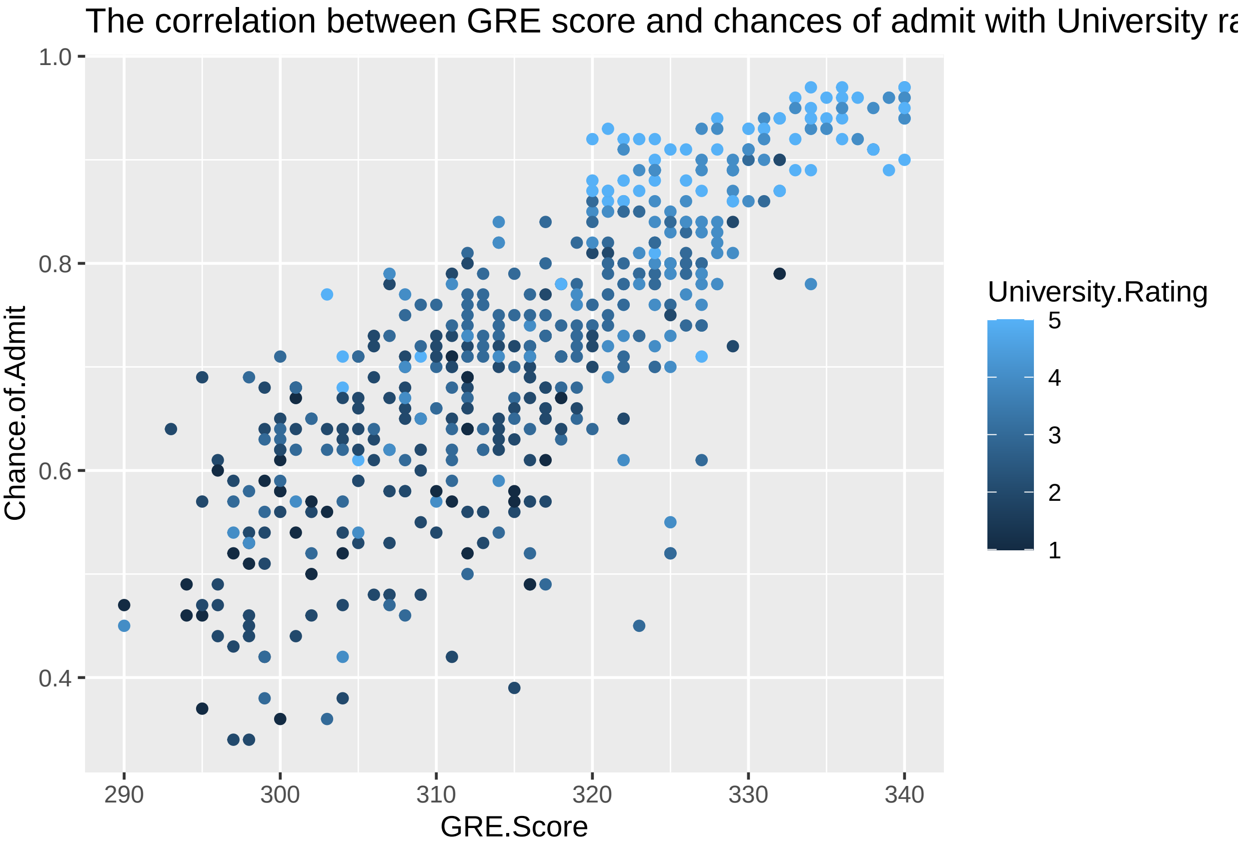
2.5 The correlation between GRE score and chances of admit with TOEFL Score column.

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=TOEFL.Score))+geom_point()+ggtitle(
  "The correlation between GRE score and chances of admit with TOEFL Score column")
```



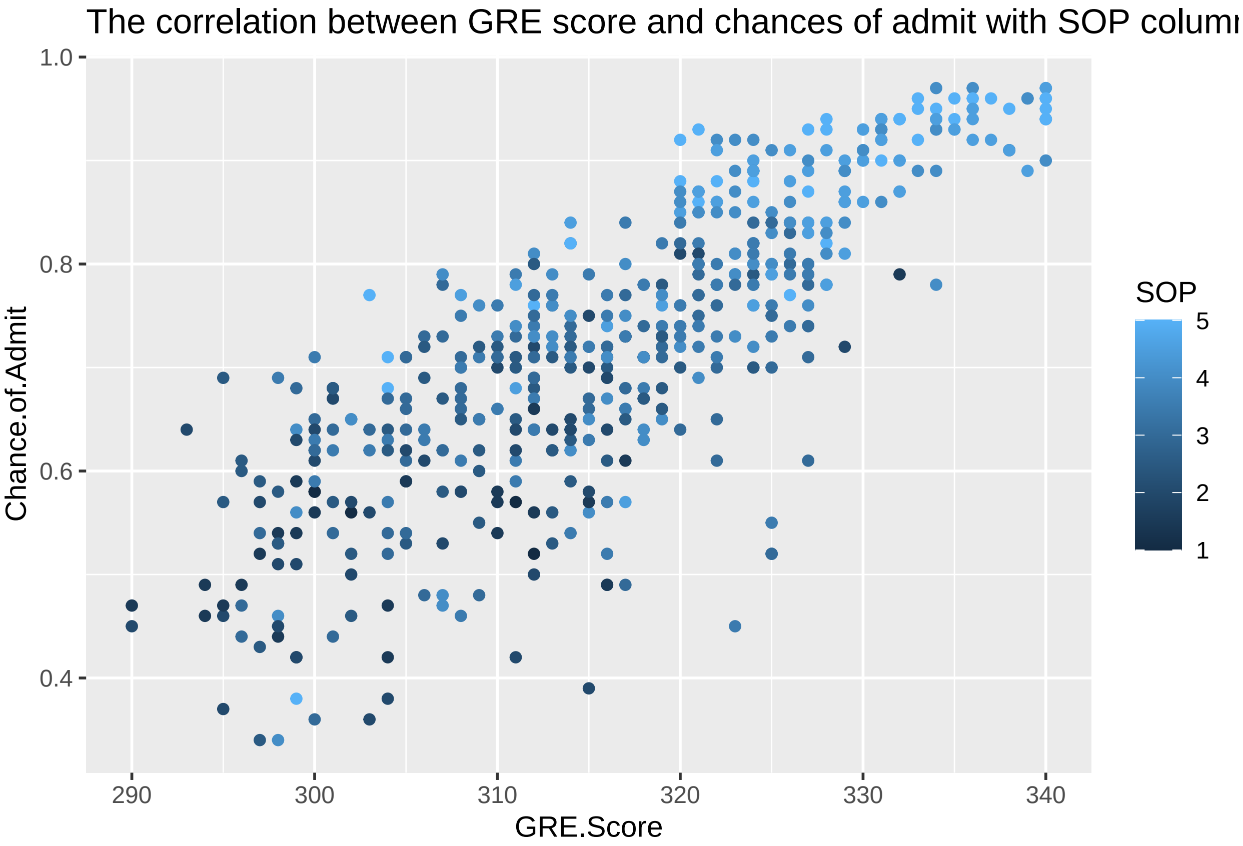
2.6 The correlation between GRE score and chances of admit with University rating column

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=University.Rating))+geom_point()+ggtitle(
  "The correlation between GRE score and chances of admit with University rating column")
```



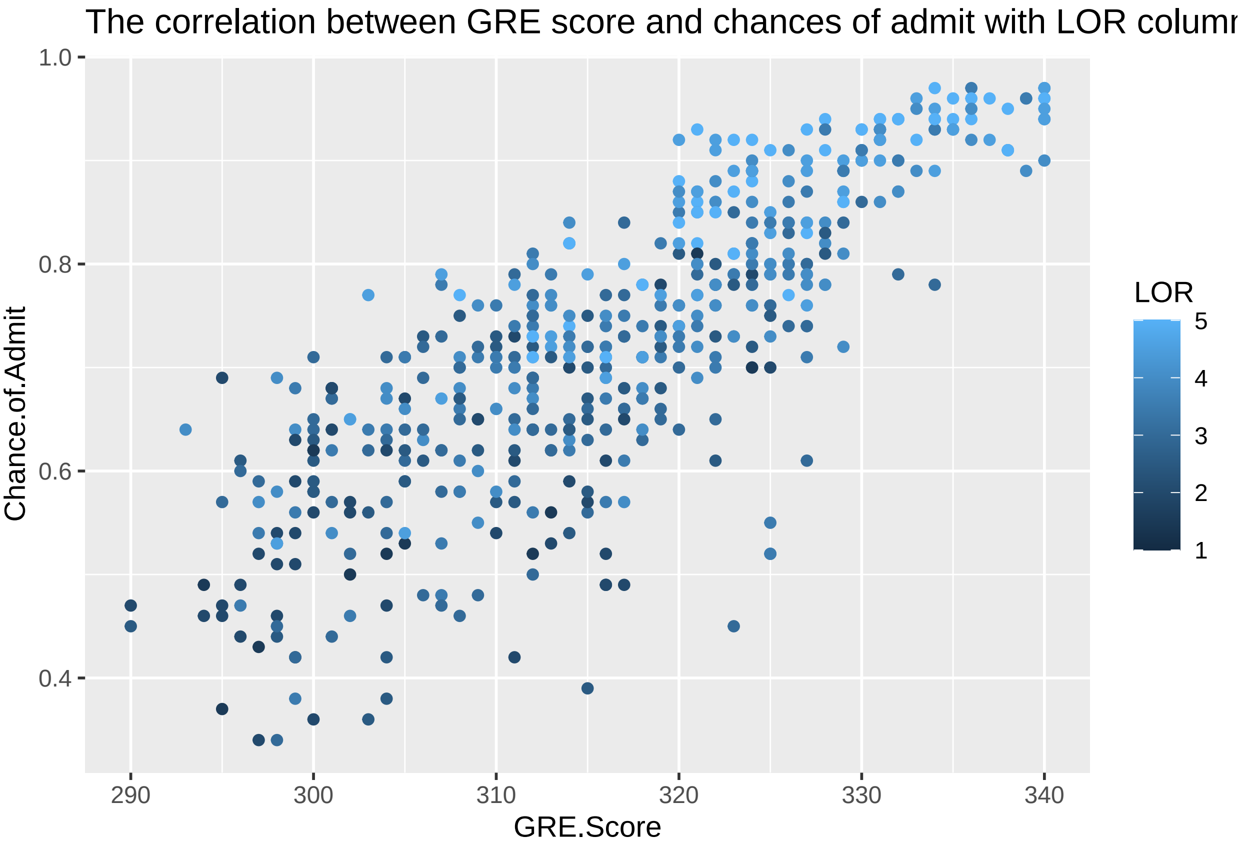
2.7 The correlation between GRE score and chances of admit with SOP column

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=SOP))+geom_point()+ggtitle(
  "The correlation between GRE score and chances of admit with SOP column")
```



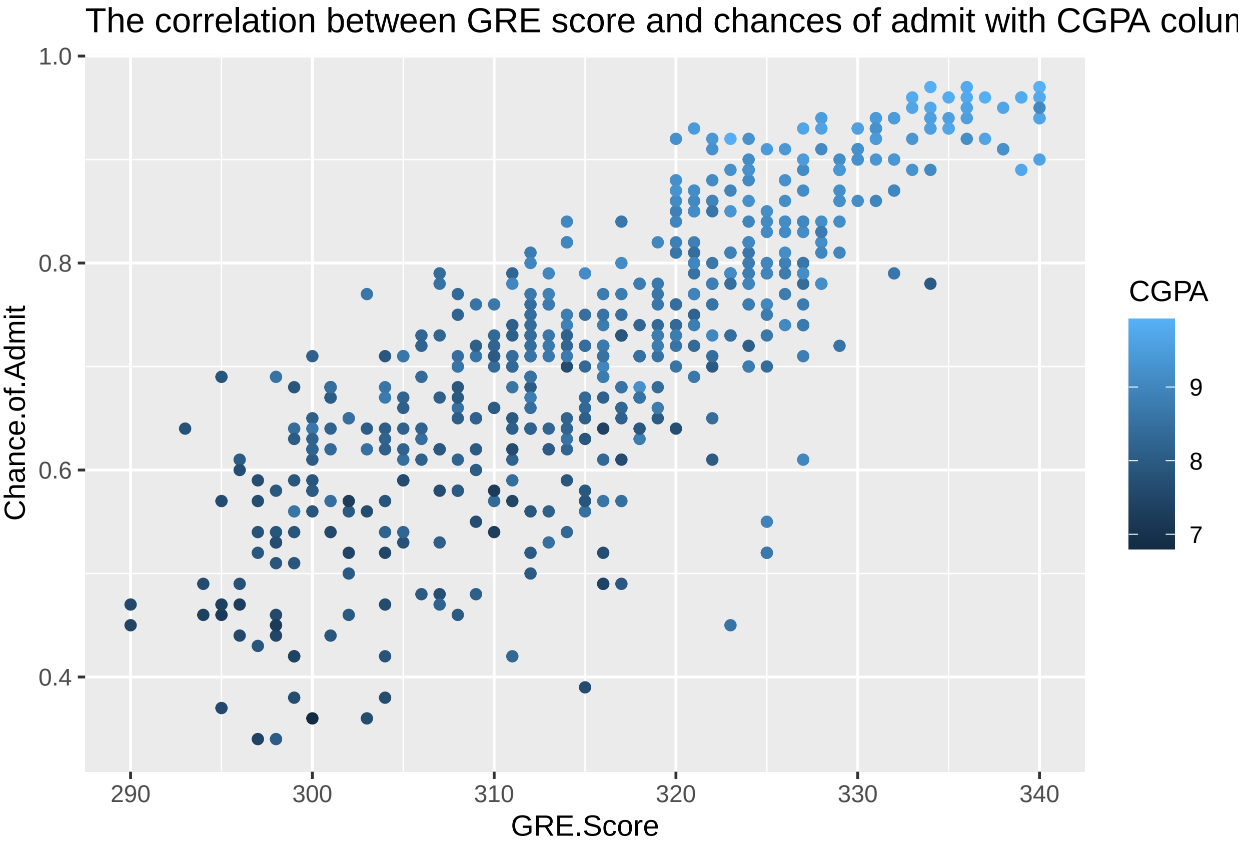
2.8 The correlation between GRE score and chances of admit with LOR column

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=LOR))+geom_point()+ggtitle(
  "The correlation between GRE score and chances of admit with LOR column")
```



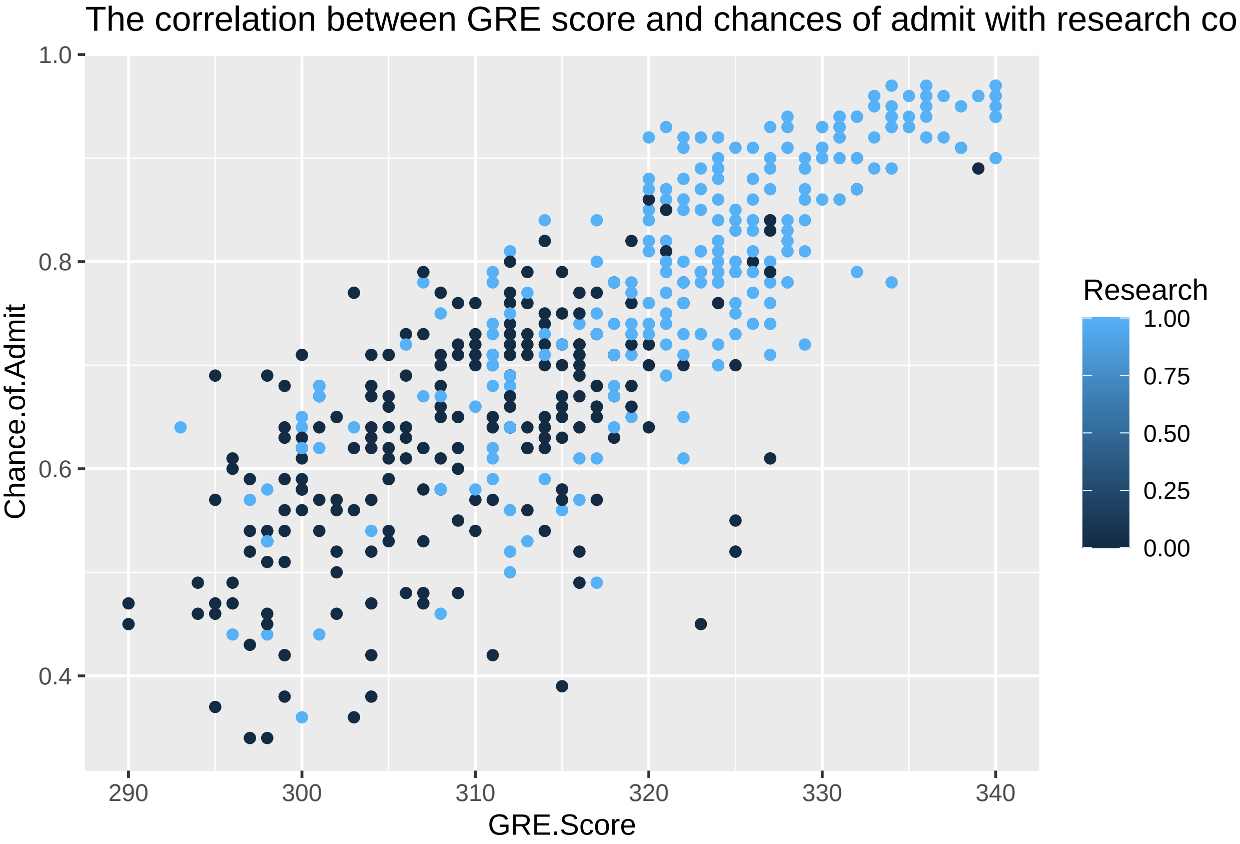
2.9 The correlation between GRE score and chances of admit with CGPA column

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=CGPA))+geom_point()+ggtitle(
  "The correlation between GRE score and chances of admit with CGPA column")
```



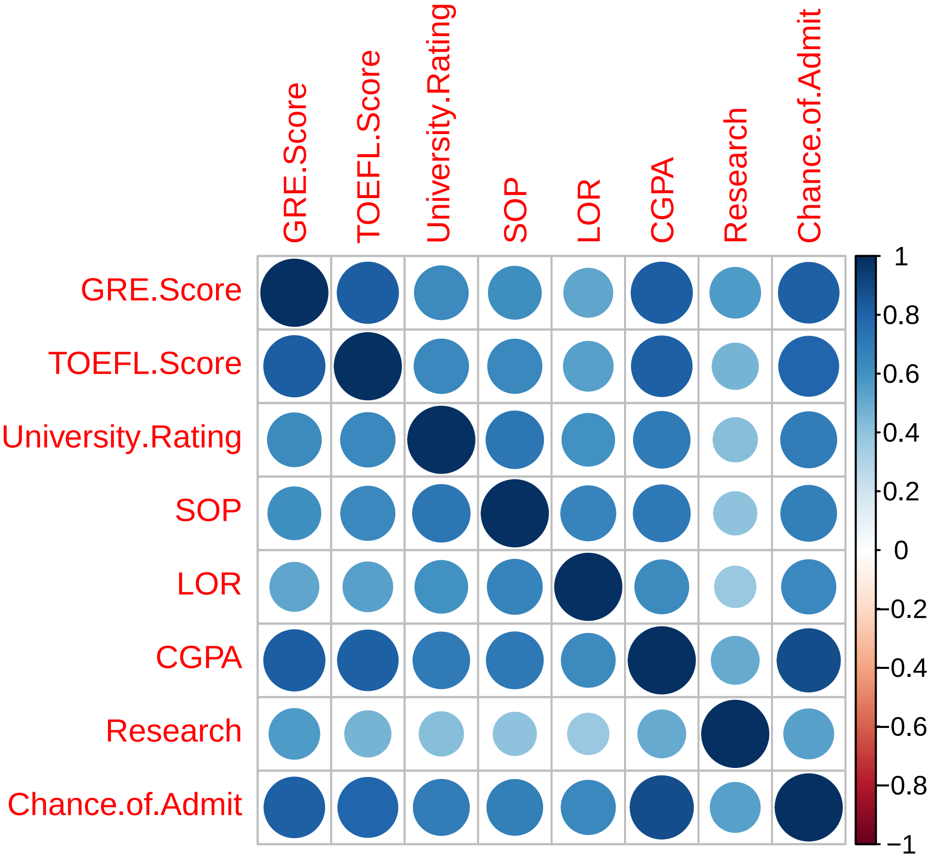
2.10 The correlation between GRE score and chances of admit with research column

```
ggplot(admission,aes(x=GRE.Score,y=Chance.of.Admit,col=Research))+geom_point()+ggtitle(
  "The correlation between GRE score and chances of admit with research column")
```



2.11 Summarize for the correlation with all different conditions.

```
library(corrplot)
admission <- admission %>% select(
  GRE.Score,TOEFL.Score,University.Rating,SOP,LOR,CGPA,Research,Chance.of.Admit)
M <- cor(admission)
corrplot(M, method = "circle")
```



3. Machine learning algorithm

Now we are focusing on three different methods, which is k-nearest neighbor, decision tree, randomforest, and linear regression models.

3.1 Data Partitioning

Generating the train and test sets are randomly splitting the data. The caret package includes the function createDataPartition that generates indexes for randomly splitting the data into training and test sets.

```
library(caret)
set.seed(1)
test_index <- createDataPartition(y = admission$Chance.of.Admit, times = 1, p = 0.5, list = FALSE)
train_set <- admission[-test_index,]
test_set <- admission[test_index,]
```

3.2 K-Nearest Neighbor

```
m_knn <- knn3(Chance.of.Admit~., data =train_set)
summary(m_knn)
```

```
##          Length Class  Mode
## learn    2      -none- list
## k        1      -none- numeric
## terms    3      terms  call
## xlevels 0      -none- list
## theDots 0      -none- list
```

```
pred <- predict(m_knn, newdata=test_set)
knn_rmse <- sqrt(mean((pred-train_set$Chance.of.Admit)^2))
rmse_results <- data_frame(method = "knn", RMSE = knn_rmse)
rmse_results
```

```
## # A tibble: 1 x 2
##   method  RMSE
##   <chr>  <dbl>
## 1 knn    0.722
```

The result is 0.722 and we can do it better.

3.3 Decision Tree

```r
library(rpart)
m_dt <- rpart(Chance.of.Admit~., data = train_set)
summary(m_dt)
```

```
## Call:
## rpart(formula = Chance.of.Admit ~ ., data = train_set)
##   n= 249 
## 
##           CP nsplit rel error    xerror       xstd
## 1 0.54212828      0 1.0000000 1.0105319 0.07730711
## 2 0.16427371      1 0.4578717 0.4830910 0.04743338
## 3 0.03846203      2 0.2935980 0.3553698 0.03662795
## 4 0.03476113      3 0.2551360 0.3263927 0.03366462
## 5 0.02368522      4 0.2203748 0.2690503 0.03155963
## 6 0.01358958      5 0.1966896 0.2733354 0.03222078
## 7 0.01203035      6 0.1831000 0.2724717 0.03232223
## 8 0.01000000      7 0.1710697 0.2607813 0.03183765
## 
## Variable importance
##              CGPA       TOEFL.Score         GRE.Score University.Rating 
##                31                17                15                13 
##               SOP               LOR          Research 
##                13                10                 1 
## 
## Node number 1: 249 observations,    complexity param=0.5421283
##   mean=0.7231325, MSE=0.01919742 
##   left son=2 (170 obs) right son=3 (79 obs)
##   Primary splits:
##       CGPA              < 8.93  to the left,  improve=0.5421283, (0 missing)
##       GRE.Score         < 319.5 to the left,  improve=0.5101993, (0 missing)
##       TOEFL.Score       < 107.5 to the left,  improve=0.4643571, (0 missing)
##       SOP               < 3.75  to the left,  improve=0.4078271, (0 missing)
##       University.Rating < 3.5   to the left,  improve=0.3894502, (0 missing)
##   Surrogate splits:
##       TOEFL.Score       < 109.5 to the left,  agree=0.867, adj=0.582, (0 split)
##       University.Rating < 3.5   to the left,  agree=0.851, adj=0.532, (0 split)
##       GRE.Score         < 320.5 to the left,  agree=0.847, adj=0.519, (0 split)
##       SOP               < 4.25  to the left,  agree=0.835, adj=0.481, (0 split)
##       LOR               < 4.25  to the left,  agree=0.803, adj=0.380, (0 split)
## 
## Node number 2: 170 observations,    complexity param=0.1642737
##   mean=0.6535882, MSE=0.01087242 
##   left son=4 (57 obs) right son=5 (113 obs)
##   Primary splits:
##       CGPA        < 8.055 to the left,  improve=0.4248495, (0 missing)
##       GRE.Score   < 304.5 to the left,  improve=0.3062640, (0 missing)
##       TOEFL.Score < 99.5  to the left,  improve=0.2787329, (0 missing)
##       LOR         < 2.75  to the left,  improve=0.2354764, (0 missing)
##       SOP         < 2.25  to the left,  improve=0.2168228, (0 missing)
##   Surrogate splits:
##       TOEFL.Score       < 101.5 to the left,  agree=0.806, adj=0.421, (0 split)
##       GRE.Score         < 300.5 to the left,  agree=0.794, adj=0.386, (0 split)
##       SOP               < 2.25  to the left,  agree=0.776, adj=0.333, (0 split)
##       University.Rating < 1.5   to the left,  agree=0.753, adj=0.263, (0 split)
##       LOR               < 2.25  to the left,  agree=0.741, adj=0.228, (0 split)
##
## Node number 3: 79 observations,    complexity param=0.03846203
##   mean=0.8727848, MSE=0.004308701
##   left son=6 (41 obs) right son=7 (38 obs)
##   Primary splits:
##       CGPA              < 9.225 to the left,  improve=0.5401333, (0 missing)
##       GRE.Score         < 328.5 to the left,  improve=0.4408685, (0 missing)
##       TOEFL.Score       < 112.5 to the left,  improve=0.2702209, (0 missing)
##       SOP               < 3.75  to the left,  improve=0.2630865, (0 missing)
##       University.Rating < 3.5   to the left,  improve=0.2569246, (0 missing)
##   Surrogate splits:
##       GRE.Score         < 328.5 to the left,  agree=0.835, adj=0.658, (0 split)
##       TOEFL.Score       < 112.5 to the left,  agree=0.759, adj=0.500, (0 split)
##       University.Rating < 4.5   to the left,  agree=0.646, adj=0.263, (0 split)
##       SOP               < 4.25  to the left,  agree=0.633, adj=0.237, (0 split)
##       LOR               < 4.25  to the left,  agree=0.608, adj=0.184, (0 split)
##
## Node number 4: 57 observations,    complexity param=0.03476113
##   mean=0.5578947, MSE=0.007802585
##   left son=8 (24 obs) right son=9 (33 obs)
##   Primary splits:
##       CGPA        < 7.695 to the left,  improve=0.3736136, (0 missing)
##       TOEFL.Score < 100.5 to the left,  improve=0.2754518, (0 missing)
##       LOR         < 2.75  to the left,  improve=0.2342456, (0 missing)
##       GRE.Score   < 299.5 to the left,  improve=0.2253933, (0 missing)
##       SOP         < 2.25  to the left,  improve=0.1363729, (0 missing)
##   Surrogate splits:
##       GRE.Score   < 298.5 to the left,  agree=0.702, adj=0.292, (0 split)
##       TOEFL.Score < 95.5  to the left,  agree=0.667, adj=0.208, (0 split)
##       SOP         < 2.25  to the left,  agree=0.649, adj=0.167, (0 split)
##       LOR         < 1.75  to the left,  agree=0.596, adj=0.042, (0 split)
##
## Node number 5: 113 observations,    complexity param=0.02368522
##   mean=0.7018584, MSE=0.005471768
##   left son=10 (58 obs) right son=11 (55 obs)
##   Primary splits:
##       CGPA              < 8.51  to the left,  improve=0.18311060, (0 missing)
##       Research          < 0.5   to the left,  improve=0.16513850, (0 missing)
##       GRE.Score         < 318.5 to the left,  improve=0.14803370, (0 missing)
##       TOEFL.Score       < 107.5 to the left,  improve=0.11517790, (0 missing)
##       University.Rating < 2.5   to the left,  improve=0.09444888, (0 missing)
##   Surrogate splits:
##       GRE.Score         < 315.5 to the left,  agree=0.717, adj=0.418, (0 split)
##       TOEFL.Score       < 106.5 to the left,  agree=0.681, adj=0.345, (0 split)
##       University.Rating < 2.5   to the left,  agree=0.611, adj=0.200, (0 split)
##       Research          < 0.5   to the left,  agree=0.611, adj=0.200, (0 split)
##       LOR               < 3.25  to the left,  agree=0.602, adj=0.182, (0 split)
##
## Node number 6: 41 observations
##   mean=0.8263415, MSE=0.003213444
##
## Node number 7: 38 observations
##   mean=0.9228947, MSE=0.0006521468
##
## Node number 8: 24 observations,    complexity param=0.01203035
##   mean=0.4945833, MSE=0.005641493
##   left son=16 (15 obs) right son=17 (9 obs)
##   Primary splits:
##       TOEFL.Score       < 100.5 to the left,  improve=0.42473200, (0 missing)
##       GRE.Score         < 304.5 to the left,  improve=0.35083860, (0 missing)
##       LOR               < 2.75  to the left,  improve=0.28565510, (0 missing)
##       CGPA              < 7.62  to the left,  improve=0.08352635, (0 missing)
##       University.Rating < 1.5   to the left,  improve=0.02978920, (0 missing)
##   Surrogate splits:
##       GRE.Score         < 304.5 to the left,  agree=0.875, adj=0.667, (0 split)
##       CGPA              < 7.62  to the left,  agree=0.833, adj=0.556, (0 split)
##       University.Rating < 2.5   to the left,  agree=0.792, adj=0.444, (0 split)
##       LOR               < 2.25  to the left,  agree=0.792, adj=0.444, (0 split)
##       Research          < 0.5   to the left,  agree=0.708, adj=0.222, (0 split)
##
## Node number 9: 33 observations
##   mean=0.6039394, MSE=0.004339027
##
## Node number 10: 58 observations
##   mean=0.6710345, MSE=0.005050654
##
## Node number 11: 55 observations,    complexity param=0.01358958
##   mean=0.7343636, MSE=0.003857322
##   left son=22 (22 obs) right son=23 (33 obs)
##   Primary splits:
##       Research          < 0.5   to the left,  improve=0.30619590, (0 missing)
##       GRE.Score         < 319.5 to the left,  improve=0.09898190, (0 missing)
##       TOEFL.Score       < 108.5 to the left,  improve=0.03748586, (0 missing)
##       University.Rating < 2.5   to the left,  improve=0.02317370, (0 missing)
##       SOP               < 3.25  to the left,  improve=0.02207897, (0 missing)
##   Surrogate splits:
##       GRE.Score         < 317.5 to the left,  agree=0.727, adj=0.318, (0 split)
##       TOEFL.Score       < 108.5 to the left,  agree=0.655, adj=0.136, (0 split)
##       University.Rating < 2.5   to the left,  agree=0.618, adj=0.045, (0 split)
##       SOP               < 2.75  to the left,  agree=0.618, adj=0.045, (0 split)
##       CGPA              < 8.86  to the right, agree=0.618, adj=0.045, (0 split)
##
## Node number 16: 15 observations
##   mean=0.4566667, MSE=0.002195556
##
## Node number 17: 9 observations
##   mean=0.5577778, MSE=0.004995062
##
## Node number 22: 22 observations
##   mean=0.6922727, MSE=0.004072107
##
## Node number 23: 33 observations
##   mean=0.7624242, MSE=0.001745638
```

```
pred<-predict(m_dt, newdata = test_set)
dt_rmse <- sqrt(mean((pred-test_set$Chance.of.Admit)^2))
```

```
rmse_results <- bind_rows(
  rmse_results, data_frame(method="Decision Tree", RMSE = dt_rmse))
rmse_results
```

```
## # A tibble: 2 x 2
##   method          RMSE
##   <chr>          <dbl>
## 1 knn           0.722
## 2 Decision Tree 0.0727
```

The result is 0.0727 and it is better than before. Then, we will use other algorithm for prediction.

3.4 Randomforest

```
library(randomForest)
m_rf <- randomForest(Chance.of.Admit~., data = train_set)
```

```
pred<-predict(m_rf,newdata = test_set)
rf_rmse <- sqrt(mean((pred-test_set$Chance.of.Admit)^2))
rmse_results <- bind_rows(
  rmse_results, data_frame(method="RandomForest", RMSE = rf_rmse))
rmse_results
```

```
## # A tibble: 3 x 2
##   method          RMSE
##   <chr>          <dbl>
## 1 knn           0.722
## 2 Decision Tree 0.0727
## 3 RandomForest  0.0654
```

The RMSE value is smaller than the last one.

3.5 Linear regression

```
m_lr <- lm(Chance.of.Admit~., data=train_set)
summary(m_lr)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ ., data = train_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22679 -0.02642  0.01002  0.03415  0.14952
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.0994629  0.1371134  -8.019 4.63e-14 ***
## GRE.Score          0.0015194  0.0006532   2.326 0.020848 *
## TOEFL.Score        0.0023865  0.0012226   1.952 0.052101 .
## University.Rating  0.0045660  0.0054386   0.840 0.401989
## SOP                0.0062199  0.0064304   0.967 0.334386
## LOR                0.0197058  0.0056992   3.458 0.000644 ***
## CGPA               0.1127974  0.0131602   8.571 1.25e-15 ***
## Research           0.0269321  0.0088807   3.033 0.002689 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0573 on 241 degrees of freedom
## Multiple R-squared:  0.8344, Adjusted R-squared:  0.8296
## F-statistic: 173.5 on 7 and 241 DF,  p-value: < 2.2e-16
```

```
pred <- predict(m_lr, newdata=test_set)
lr_RMSE <- sqrt(mean((pred-test_set$Chance.of.Admit)^2))
rmse_results <- bind_rows(
  rmse_results, data_frame(method = "Linear regression", RMSE = lr_RMSE))
rmse_results
```

```
## # A tibble: 4 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 knn               0.722
## 2 Decision Tree     0.0727
## 3 RandomForest      0.0654
## 4 Linear regression 0.0631
```

The RMSE result is 0.0631 and we believe that it is the best result compare with the others.

4. Conclusion

Using data from Mohan S Acharya data-set sourced from Kaggle several predictors or covariates were utilized to predict students in shortlisting universities with their profiles. After the use of several models the highest accuracy of 0.0631 was established by a Linear regression model with predictors TOFEL score, University rating, SOP, LOR, and CGPA.