```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

import statsmodels.api as sm

from statsmodels.stats.outliers_influence import
variance_inflation_factor

from sklearn.linear_model import LinearRegression

from statsmodels.graphics.gofplots import qqplot

import warnings
warnings.filterwarnings("ignore")
```

#Define Problem Statement and perform Exploratory Data Analysis

Why this case study?

From company's perspective:

● Jamboree is a renowned educational institution that has successfully assisted numerous students in gaining admission to top colleges abroad. With their proven problem-solving methods, they have helped students achieve exceptional scores on exams like GMAT, GRE, and SAT with minimal effort.

● To further support students, Jamboree has recently introduced a new feature on their website. This feature enables students to assess their probability of admission to Ivy League colleges, considering the unique perspective of Indian applicants.

● By conducting a thorough analysis, we can assist Jamboree in understanding the crucial factors impacting graduate admissions and their interrelationships. Additionally, we can provide predictive insights to determine an individual's admission chances based on various variables.

From learner's perspective:

● Solving this business case holds immense importance for aspiring data scientists and ML engineers.

● Building predictive models using machine learning is widely popular among the data scientists/ML engineers. By working through this case study, individuals gain hands-on experience and practical skills in the field.

● Additionally, it will enhance one's ability to communicate with the stakeholders involved in data-related projects and help the organization take better, data-driven decisions.

Assuming you're a data scientist/ML engineer hired by Jamboree, your primary objective is to analyze the given dataset and derive valuable insights from it. Additionally, utilize the dataset to construct a predictive model capable of estimating an applicant's likelihood of admission based on the available features.

We are supposed to find the importance of predictor variables and how the model can be improved and how the business can benefit from it.

```
data = pd.read_csv('./Admission_Predict_Ver1.1.csv')

data.head()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 500,\n  \"fields\":
[\n    {\n      \"column\": \"Serial No.\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 144,\n        \"min\": 1,\n
\"max\": 500,\n        \"num_unique_values\": 500,\n
\"samples\": [\n          362,\n          74,\n          375\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"GRE
Score\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 11,\n        \"min\": 290,\n        \"max\": 340,\n
\"num_unique_values\": 49,\n        \"samples\": [\n          307,\n
335,\n          297\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"TOEFL Score\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 6,\n        \"min\": 92,\n
\"max\": 120,\n        \"num_unique_values\": 29,\n
\"samples\": [\n          94,\n          119,\n          112\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"University Rating\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1,\n        \"min\": 1,\n
\"max\": 5,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          3,\n          1,\n          2\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SOP\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.9910036207566072,\n
\"min\": 1.0,\n        \"max\": 5.0,\n        \"num_unique_values\":
9,\n        \"samples\": [\n          1.0,\n          4.0,\n
5.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"LOR
\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.9254495738978193,\n        \"min\": 1.0,\n        \"max\":
5.0,\n        \"num_unique_values\": 9,\n        \"samples\": [\n
5.0,\n          3.5,\n          1.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"CGPA\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.6048128003332054,\n
\"min\": 6.8,\n        \"max\": 9.92,\n        \"num_unique_values\":
184,\n        \"samples\": [\n          9.6,\n          8.9,\n
```

8.24\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Research\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0,\n   1\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Chance of Admit \",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.14114040395030228,\n   \"min\": 0.34,\n   \"max\": 0.97,\n   \"num_unique_values\": 61,\n   \"samples\": [\n   0.92,\n   0.9\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"data"}

```
data.columns
```

```
Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating',
'SOP',
       'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
      dtype='object')
```

```
data.rename(columns = lambda x : x.strip(), inplace = True)
```

```
data.columns
```

```
Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating',
'SOP',
       'LOR', 'CGPA', 'Research', 'Chance of Admit'],
      dtype='object')
```

```
data.shape
```

```
(500, 9)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Serial No.         500 non-null    int64
 1   GRE Score          500 non-null    int64
 2   TOEFL Score        500 non-null    int64
 3   University Rating  500 non-null    int64
 4   SOP                500 non-null    float64
 5   LOR                500 non-null    float64
 6   CGPA               500 non-null    float64
 7   Research           500 non-null    int64
 8   Chance of Admit    500 non-null    float64
```

```
dtypes: float64(4), int64(5)
memory usage: 35.3 KB
```

```
data.duplicated().sum()
```

```
0
```

```
data.describe()
```

{"summary":"{\n \"name\": \"data\",\n \"rows\": 8,\n \"fields\": [\n   {\n   \"column\": \"Serial No.\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 179.8977277873755,\n   \"min\": 1.0,\n   \"max\": 500.0,\n   \"num_unique_values\": 6,\n   \"samples\": [\n   500.0,\n   250.5,\n   375.25\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 134.31959598717793,\n   \"min\": 11.2951483723547,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   316.472,\n   317.0,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"TOEFL Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 148.54698537663884,\n   \"min\": 6.081867659564528,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   107.192,\n   107.0,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"University Rating\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 175.8093363236959,\n   \"min\": 1.0,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   3.114,\n   3.0,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 175.75364204315028,\n   \"min\": 0.9910036207566072,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   3.374,\n   3.5,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"LOR\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 175.72621272918164,\n   \"min\": 0.9254495738978193,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   3.484,\n   3.5,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"CGPA\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 174.19317432229437,\n   \"min\": 0.6048128003332054,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   8.576439999999998,\n   8.56,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n

\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Research\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 176.57228090801308,\n   \"min\": 0.0,\n   \"max\": 500.0,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   0.56,\n   1.0,\n   0.49688407860903566\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Chance of Admit\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 176.5575452133998,\n   \"min\": 0.14114040395030228,\n   \"max\": 500.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   0.72174,\n   0.72,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
data.isna().sum()
```

```
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

```
data.drop(columns = ['Serial No.'], inplace = True)
```

```
data.head()
```

{"summary":"{\n \"name\": \"data\",\n \"rows\": 500,\n \"fields\": [\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 11,\n   \"min\": 290,\n   \"max\": 340,\n   \"num_unique_values\": 49,\n   \"samples\": [\n   307,\n   335,\n   297\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"TOEFL Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 6,\n   \"min\": 92,\n   \"max\": 120,\n   \"num_unique_values\": 29,\n   \"samples\": [\n   94,\n   119,\n   112\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"University Rating\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1,\n   \"min\": 1,\n   \"max\": 5,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   3,\n   1,\n   2\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n

\"dtype\": \"number\",\n   \"std\": 0.9910036207566072,\n
\"min\": 1.0,\n   \"max\": 5.0,\n   \"num_unique_values\":
9,\n   \"samples\": [\n   1.0,\n   4.0,\n
5.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"LOR\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0.9254495738978193,\n   \"min\": 1.0,\n   \"max\":
5.0,\n   \"num_unique_values\": 9,\n   \"samples\": [\n
5.0,\n   3.5,\n   1.5\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"CGPA\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.6048128003332054,\n
\"min\": 6.8,\n   \"max\": 9.92,\n   \"num_unique_values\":
184,\n   \"samples\": [\n   9.6,\n   8.9,\n
8.24\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Research\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0,\n   \"min\": 0,\n
\"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   0,\n   1\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"Chance of Admit\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.14114040395030228,\n
\"min\": 0.34,\n   \"max\": 0.97,\n   \"num_unique_values\":
61,\n   \"samples\": [\n   0.92,\n   0.9\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"data"}

```
data.nunique()
```

```
GRE Score            49
TOEFL Score          29
University Rating     5
SOP                   9
LOR                   9
CGPA                184
Research              2
Chance of Admit      61
dtype: int64
```

```
for col in data.columns:
  print(col, '\n', data[col].value_counts(), '\n')
```

```
GRE Score
 GRE Score
312    24
324    23
316    18
321    17
```

```
322    17
327    17
311    16
320    16
314    16
317    15
325    15
315    13
308    13
323    13
326    12
319    12
313    12
304    12
300    12
318    12
305    11
301    11
310    11
307    10
329    10
299    10
298    10
331     9
340     9
328     9
309     9
334     8
332     8
330     8
306     7
302     7
297     6
296     5
295     5
336     5
303     5
338     4
335     4
333     4
339     3
337     2
290     2
294     2
293     1
Name: count, dtype: int64

TOEFL Score
 TOEFL Score
```

```
110    44
105    37
104    29
107    28
106    28
112    28
103    25
100    24
102    24
99     23
101    20
111    20
108    19
113    19
109    19
114    18
116    16
115    11
118    10
98     10
119    10
120     9
117     8
97      7
96      6
95      3
93      2
94      2
92      1
Name: count, dtype: int64

University Rating
 University Rating
3    162
2    126
4    105
5     73
1     34
Name: count, dtype: int64

SOP
 SOP
4.0    89
3.5    88
3.0    80
2.5    64
4.5    63
2.0    43
5.0    42
```

```
1.5    25
1.0     6
Name: count, dtype: int64

LOR
 LOR
3.0    99
4.0    94
3.5    86
4.5    63
2.5    50
5.0    50
2.0    46
1.5    11
1.0     1
Name: count, dtype: int64

CGPA
 CGPA
8.76    9
8.00    9
8.12    7
8.45    7
8.54    7
       ..
9.92    1
9.35    1
8.71    1
9.32    1
7.69    1
Name: count, Length: 184, dtype: int64

Research
 Research
1    280
0    220
Name: count, dtype: int64

Chance of Admit
 Chance of Admit
0.71    23
0.64    19
0.73    18
0.72    16
0.79    16
        ..
0.38     2
0.36     2
0.43     1
0.39     1
```

```
0.37     1
Name: count, Length: 61, dtype: int64
```

Outlier Detection using IQR method

```
#Calculating few more statistical measures such as 'Range', 'IQR',
'Lower Whisker' and 'Upper Whisker'

descriptive_stats = data.describe()
descriptive_stats =
descriptive_stats.reindex(descriptive_stats.index.values.tolist()+
['Range', 'IQR', 'Lower Whisker', 'Upper Whisker'])

for col in descriptive_stats.columns:
  descriptive_stats.loc['Range'][col] = descriptive_stats.loc['max']
[col] - descriptive_stats.loc['min'][col]
  descriptive_stats.loc['IQR'][col] = descriptive_stats.loc['75%']
[col] - descriptive_stats.loc['25%'][col]
  descriptive_stats.loc['Lower Whisker'][col] =
descriptive_stats.loc['25%'][col] - (1.5 *
descriptive_stats.loc['IQR'][col])
  descriptive_stats.loc['Upper Whisker'][col] =
descriptive_stats.loc['75%'][col] + (1.5 *
descriptive_stats.loc['IQR'][col])

descriptive_stats
```


{"summary":"{\n \"name\": \"descriptive_stats\",\n \"rows\": 12,\n \"fields\": [\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 151.2235611108601,\n   \"min\": 11.2951483723547,\n   \"max\": 500.0,\n   \"num_unique_values\": 12,\n   \"samples\": [\n   282.5,\n   17.0,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"TOEFL Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 128.08188701172082,\n   \"min\": 6.081867659564528,\n   \"max\": 500.0,\n   \"num_unique_values\": 12,\n   \"samples\": [\n   89.5,\n   9.0,\n   500.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"University Rating\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 143.5322134558346,\n   \"min\": -1.0,\n   \"max\": 500.0,\n   \"num_unique_values\": 10,\n   \"samples\": [\n   -1.0,\n   3.114,\n   3.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 143.49929770945212,\n   \"min\": 0.25,\n   \"max\": 500.0,\n   \"num_unique_values\": 11,\n   \"samples\": [\n

3.5,\n   500.0,\n   0.25\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"LOR\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 143.4825629546811,\n
\"min\": 0.9254495738978193,\n   \"max\": 500.0,\n
\"num_unique_values\": 10,\n   \"samples\": [\n   1.5,\n
3.484,\n   3.5\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"CGPA\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 142.46524631259436,\n   \"min\": 0.6048128003332054,\n
\"max\": 500.0,\n   \"num_unique_values\": 12,\n
\"samples\": [\n   6.7587500000000045,\n
0.9124999999999979,\n   500.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Research\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
144.15537814187454,\n   \"min\": -1.5,\n   \"max\": 500.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n   500.0,\n
0.56,\n   -1.5\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Chance of Admit\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 144.16433400704474,\n   \"min\":
0.14114040395030228,\n   \"max\": 500.0,\n
\"num_unique_values\": 12,\n   \"samples\": [\n
0.3450000000000001,\n   0.18999999999999995,\n   500.0\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\
n}","type":"dataframe","variable_name":"descriptive_stats"}

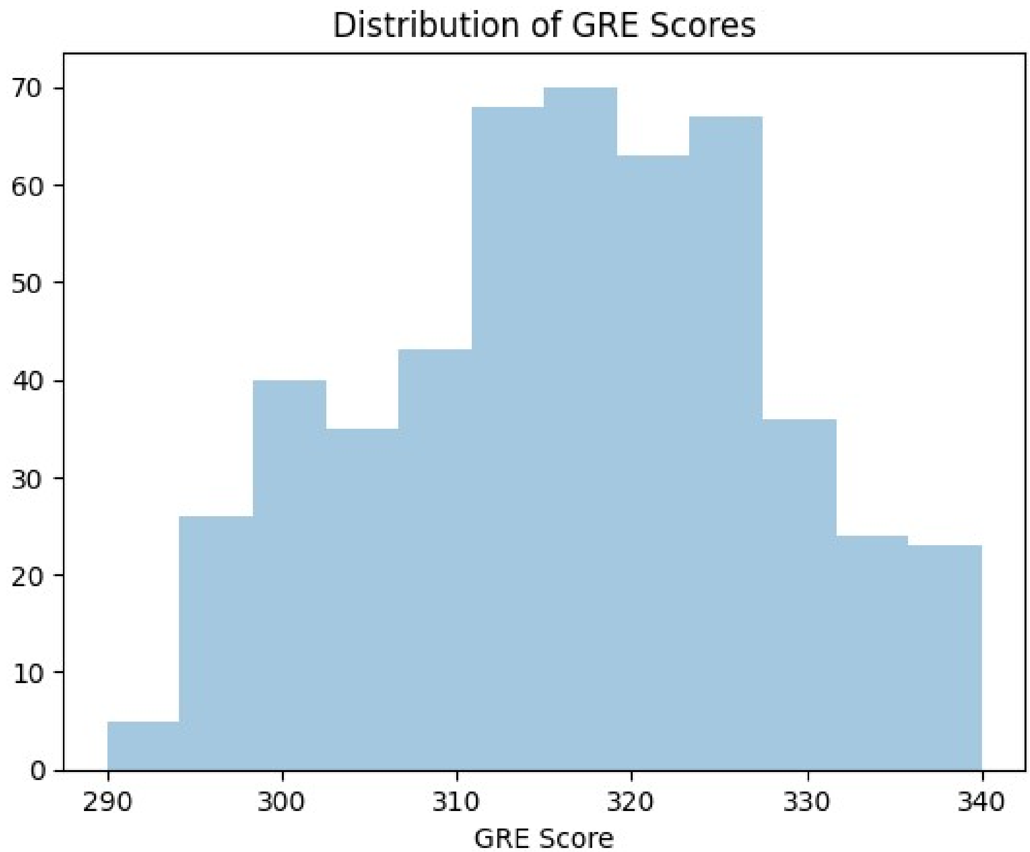

```
categorical_variables = ['University Rating', 'SOP', 'LOR',
'Research']

for col in descriptive_stats.columns:
  if col not in categorical_variables:
    print(col, ':', data[(data[col] < descriptive_stats.loc['Lower
Whisker'][col]) | (data[col] > descriptive_stats.loc['Upper Whisker']
[col])][col].count())
```

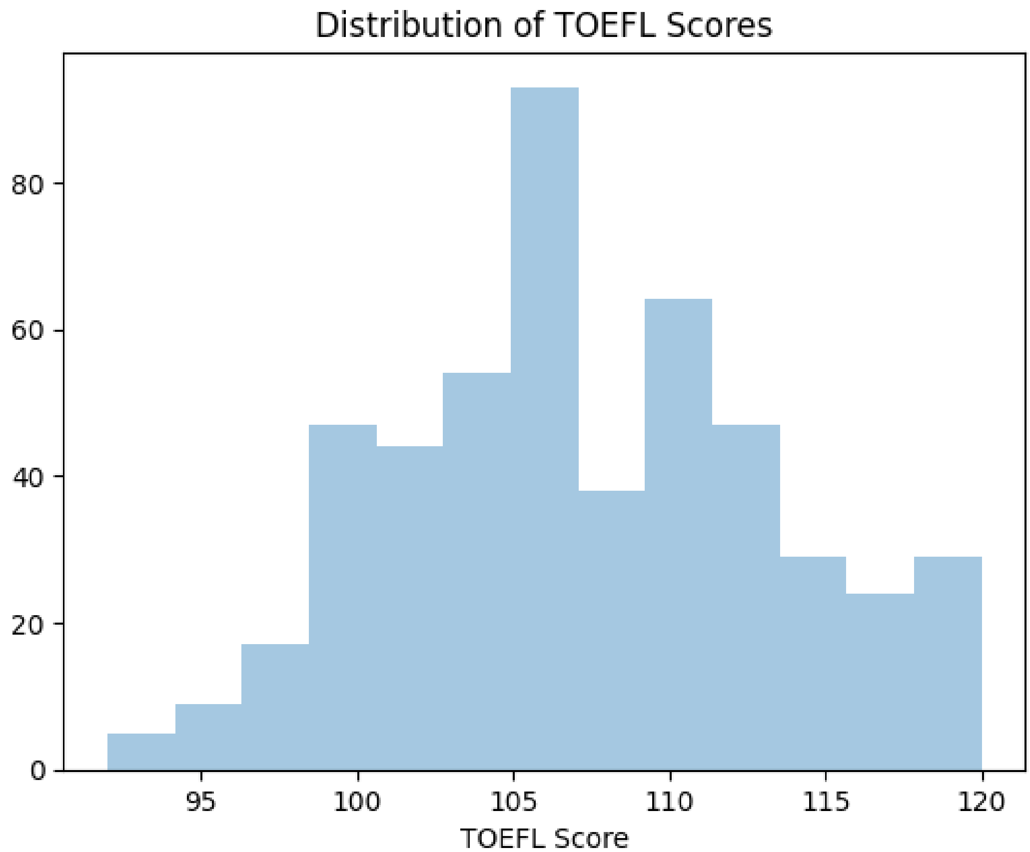
```
GRE Score : 0
TOEFL Score : 0
CGPA : 0
Chance of Admit : 2
```

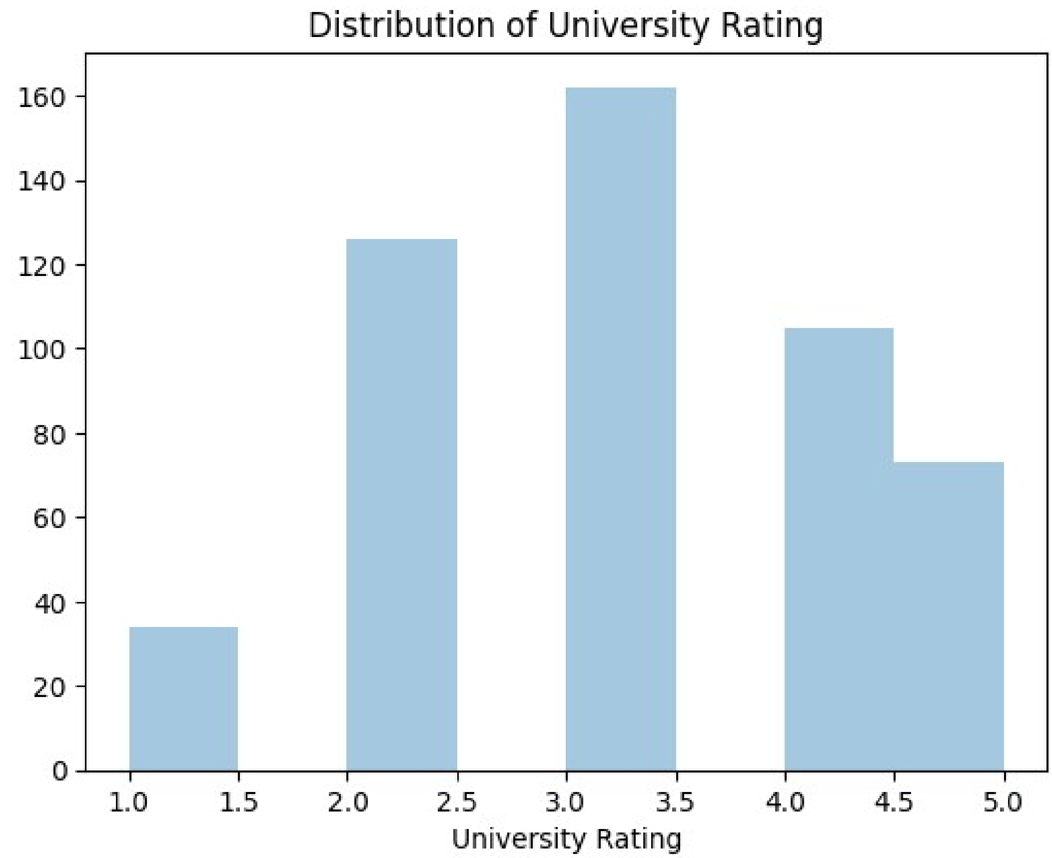
##Univariate Analysis

Lets see the distribution of the variables of graduate applicants.

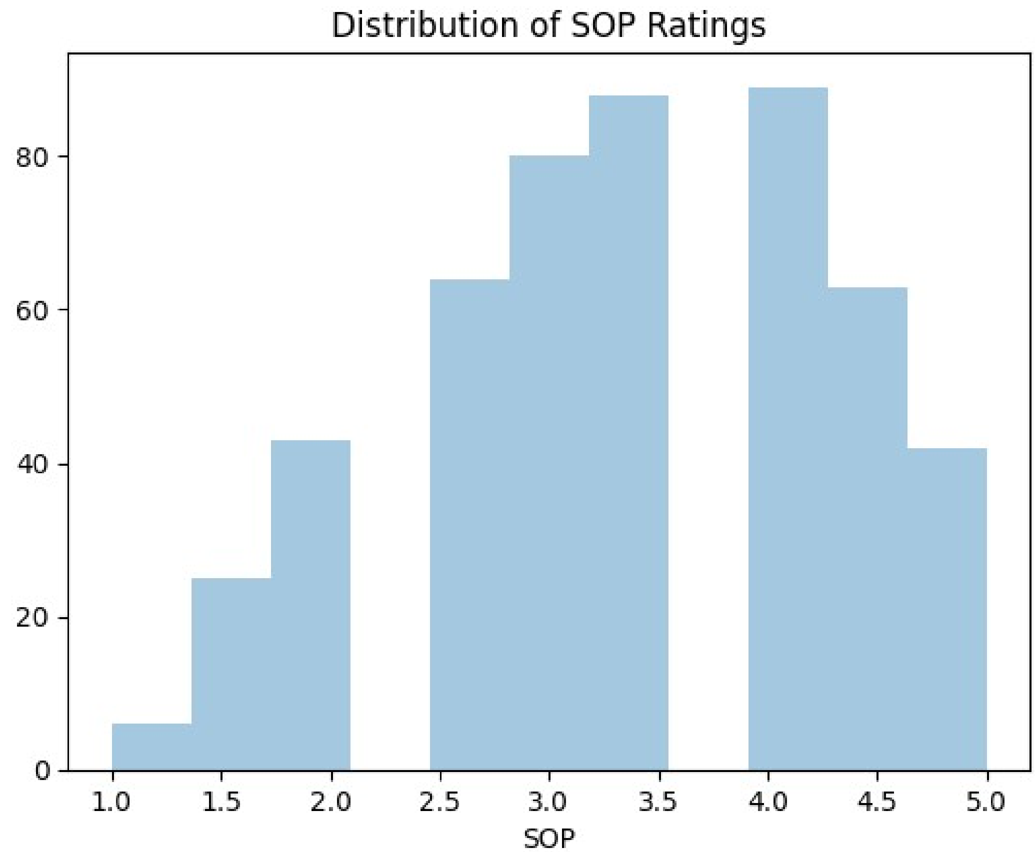
```
fig = sns.distplot(data['GRE Score'], kde=False)
plt.title("Distribution of GRE Scores")
plt.show()

fig = sns.distplot(data['TOEFL Score'], kde=False)
plt.title("Distribution of TOEFL Scores")
plt.show()

fig = sns.distplot(data['University Rating'], kde=False)
plt.title("Distribution of University Rating")
plt.show()

fig = sns.distplot(data['SOP'], kde=False)
plt.title("Distribution of SOP Ratings")
plt.show()

fig = sns.distplot(data['LOR'], kde=False)
plt.title("Distribution of LOR Ratings")
plt.show()

fig = sns.distplot(data['CGPA'], kde=False)
plt.title("Distribution of CGPA")
plt.show()

plt.show()
```

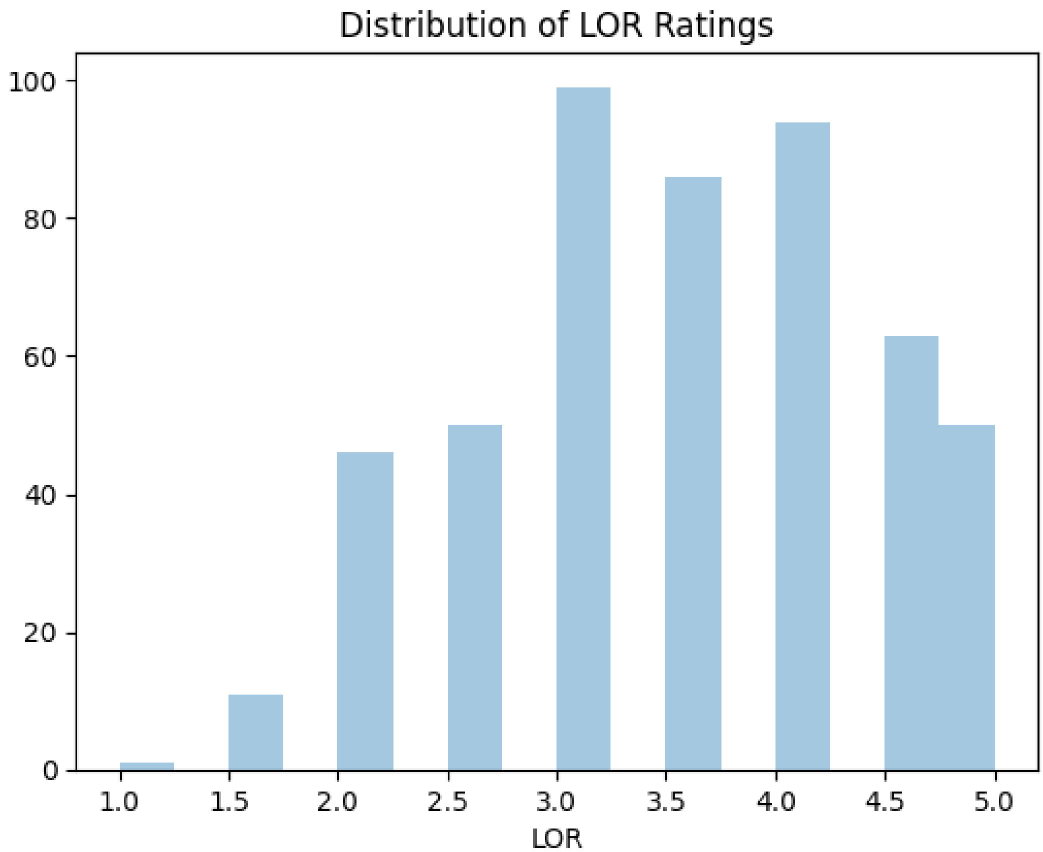

Distribution of GRE Scores
0
10
20
30
40
50
60
70
290
300
310
320
330
340
GRE Score


Distribution of TOEFL Scores
0
20
40
60
80
95
100
105
110
115
120
TOEFL Score

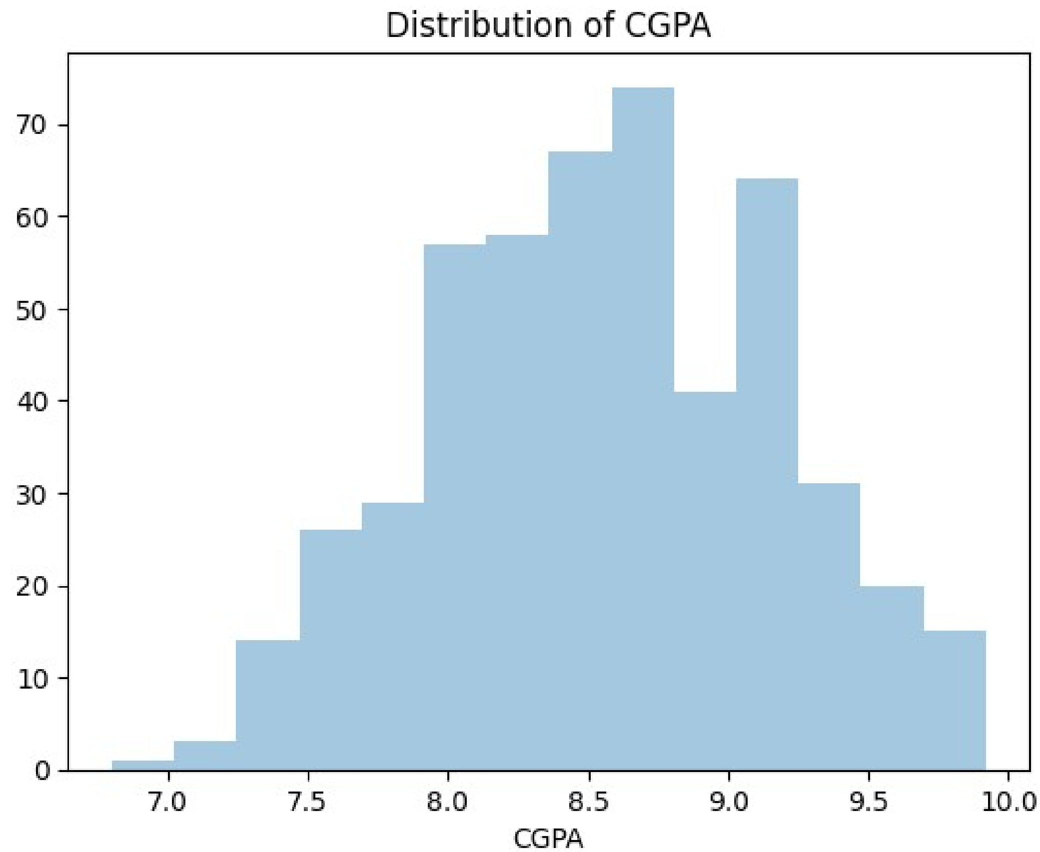


Distribution of University Rating
160
140
120
100
80
60
40
20
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
University Rating


Distribution of SOP Ratings
0
20
40
60
80
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
SOP



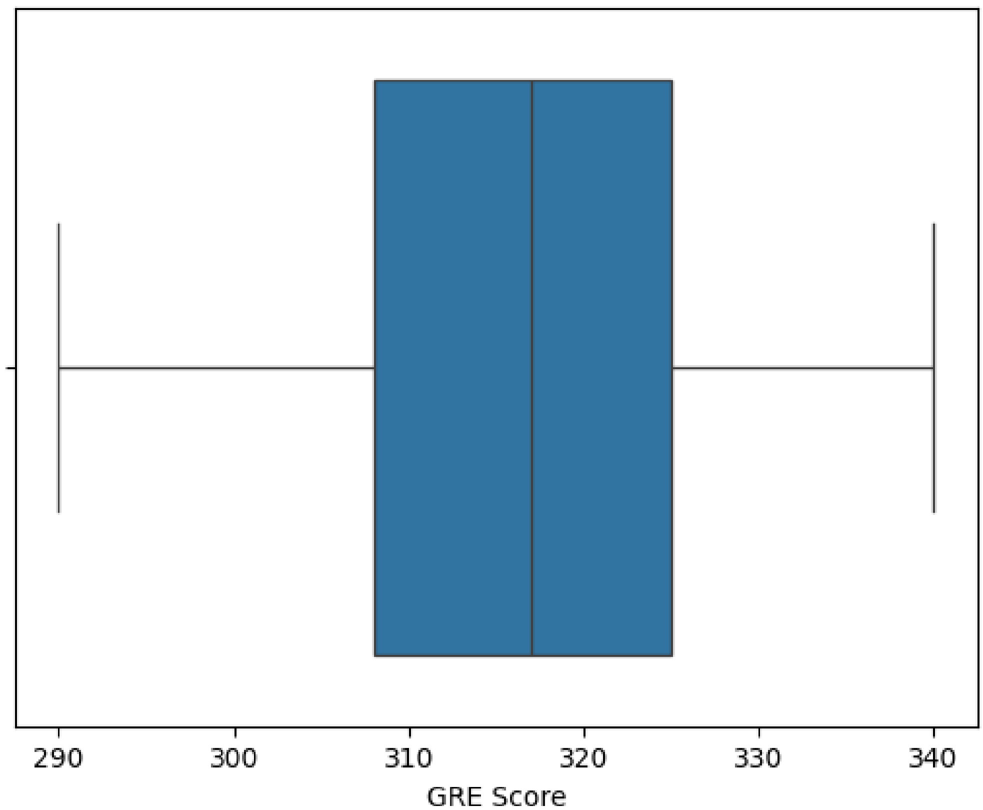
Distribution of LOR Ratings
100
80
60
40
20
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
LOR

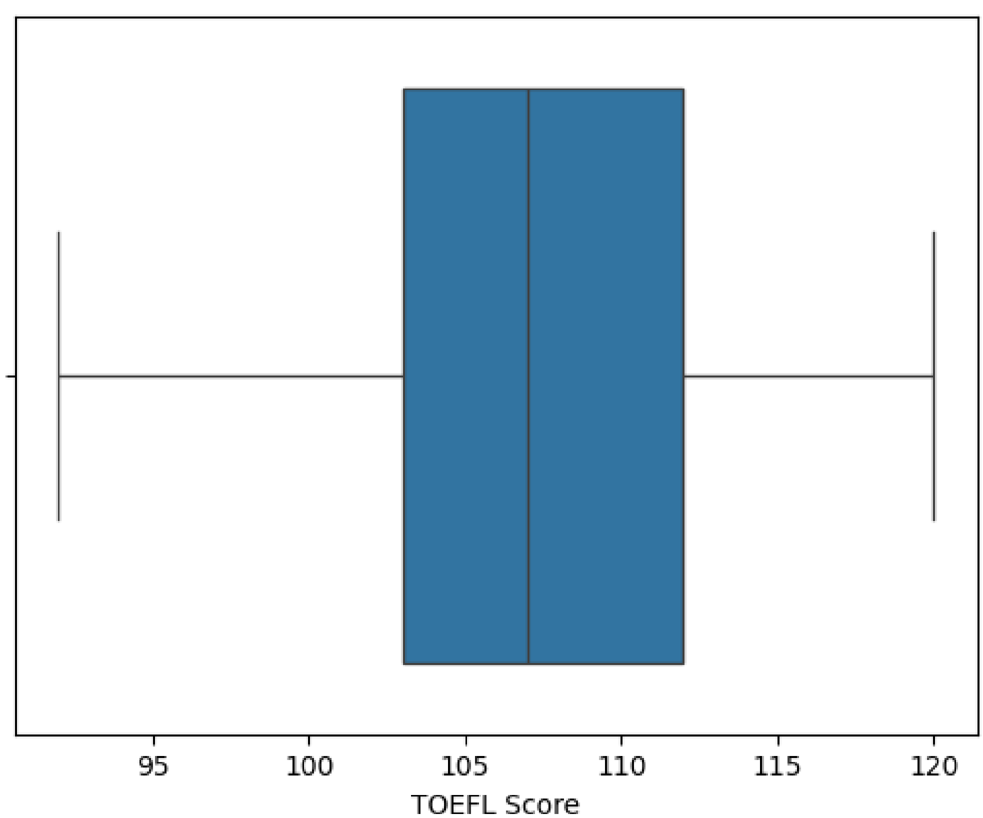


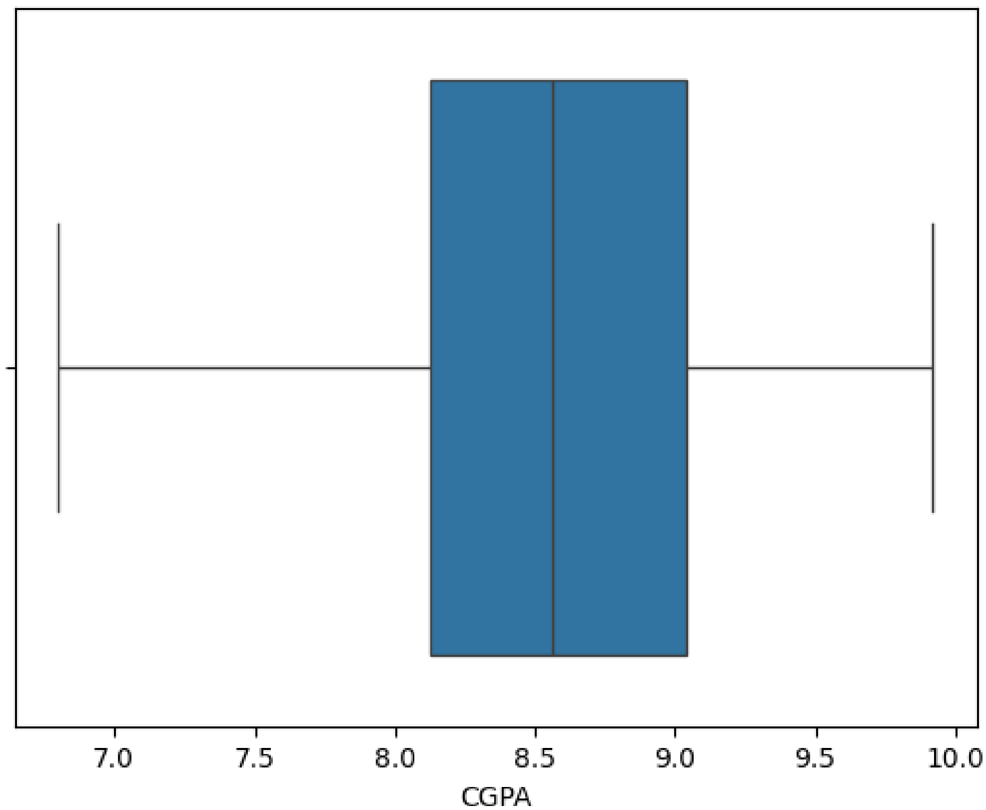
It is clear from the distributions, students with varied merit apply for the university.

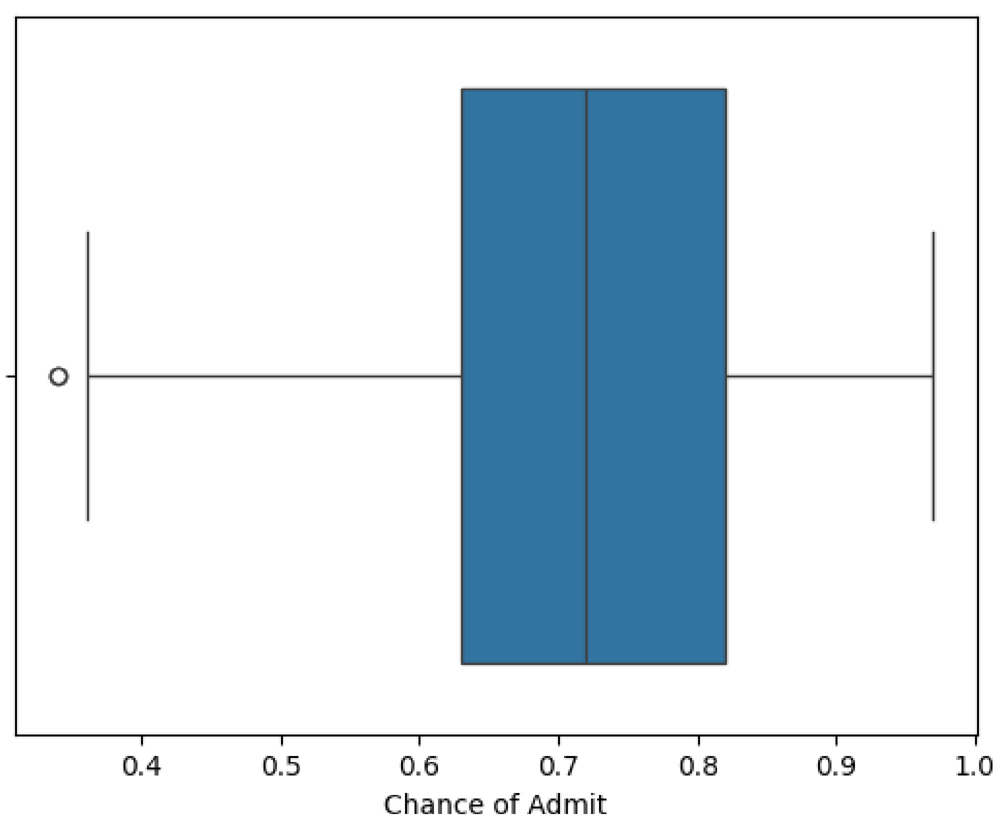
###Continuous Variables

```
categorical_variables = ['University Rating', 'SOP', 'LOR',
'Research']

for col in data.columns:
  if col not in categorical_variables:
    sns.boxplot(data = data, x = col)
    plt.show()
```

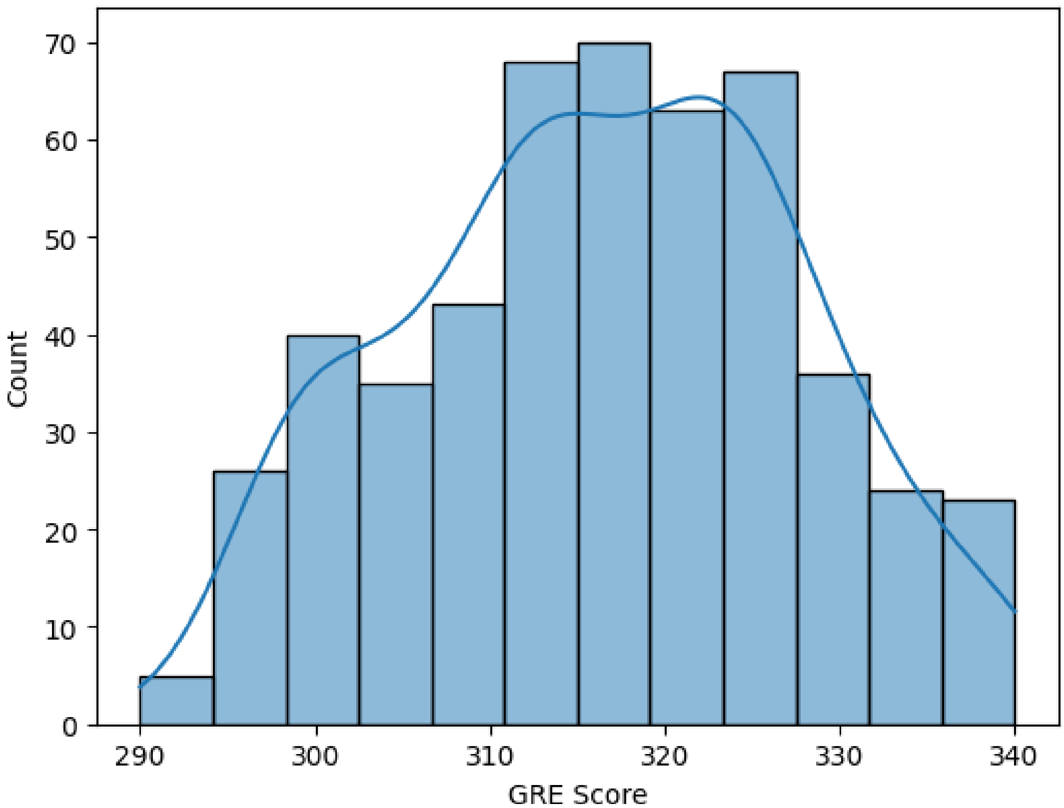

290
300
310
320
330
340
GRE Score

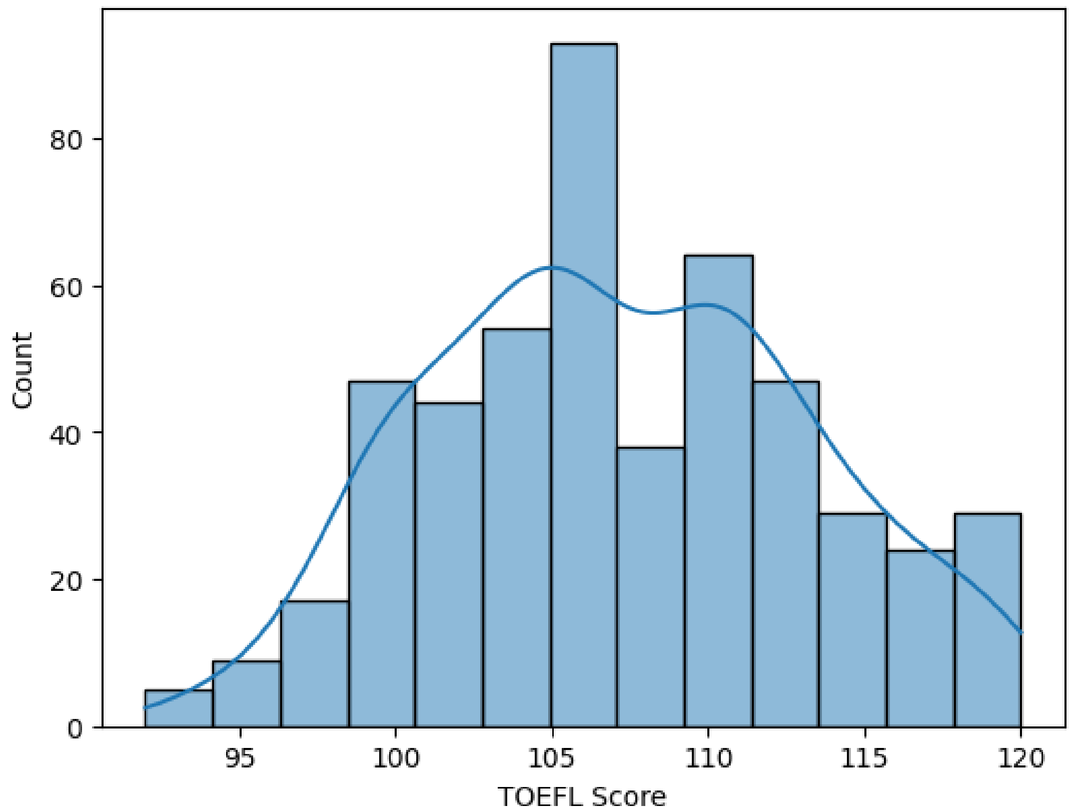

95
100
105
110
115
120
TOEFL Score

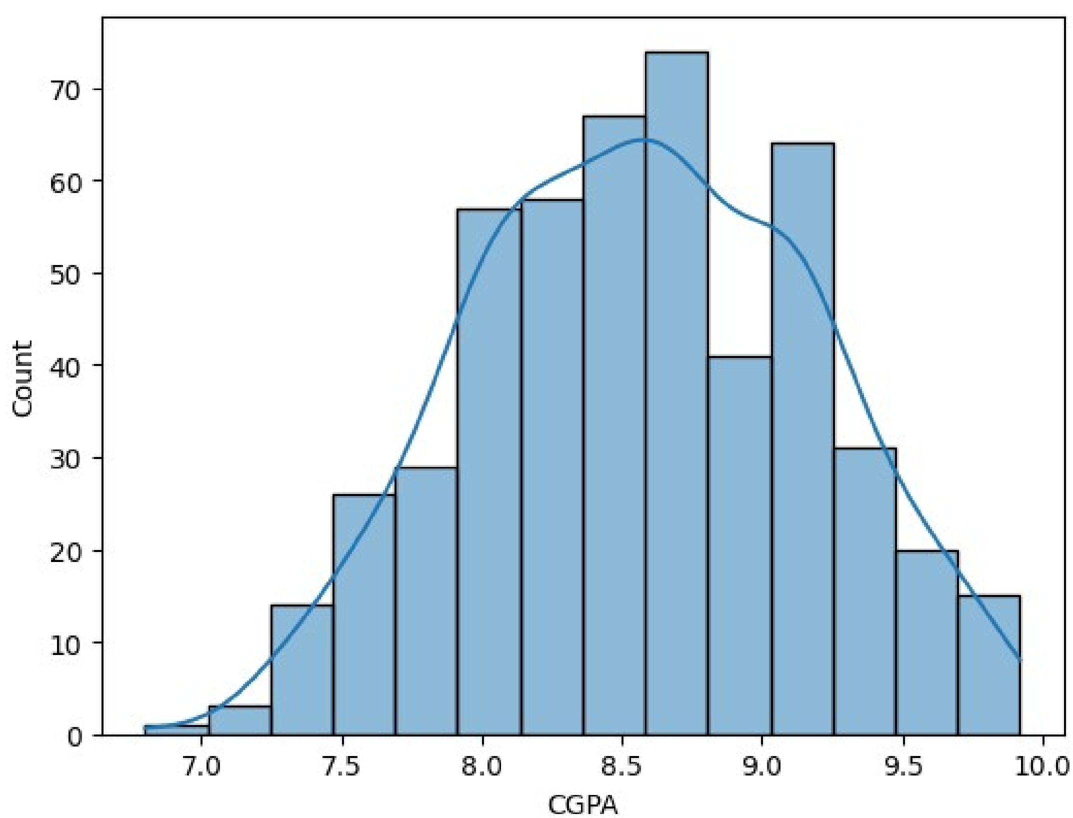

7.0
7.5
8.0
8.5
9.0
9.5
10.0
CGPA

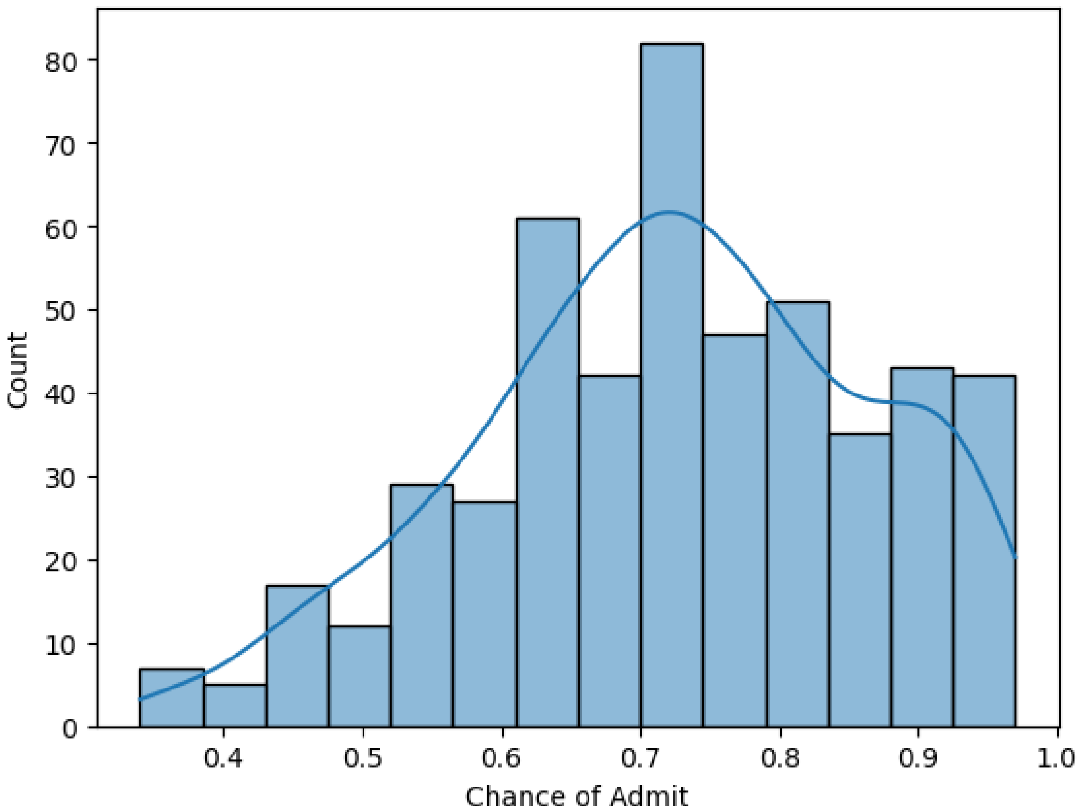

0.4
0.5
0.6
0.7
0.8
0.9
1.0
Chance of Admit

```
for col in data.columns:
  if col not in categorical_variables:
    sns.histplot(data = data, x = col, kde = True)
    plt.show()
```



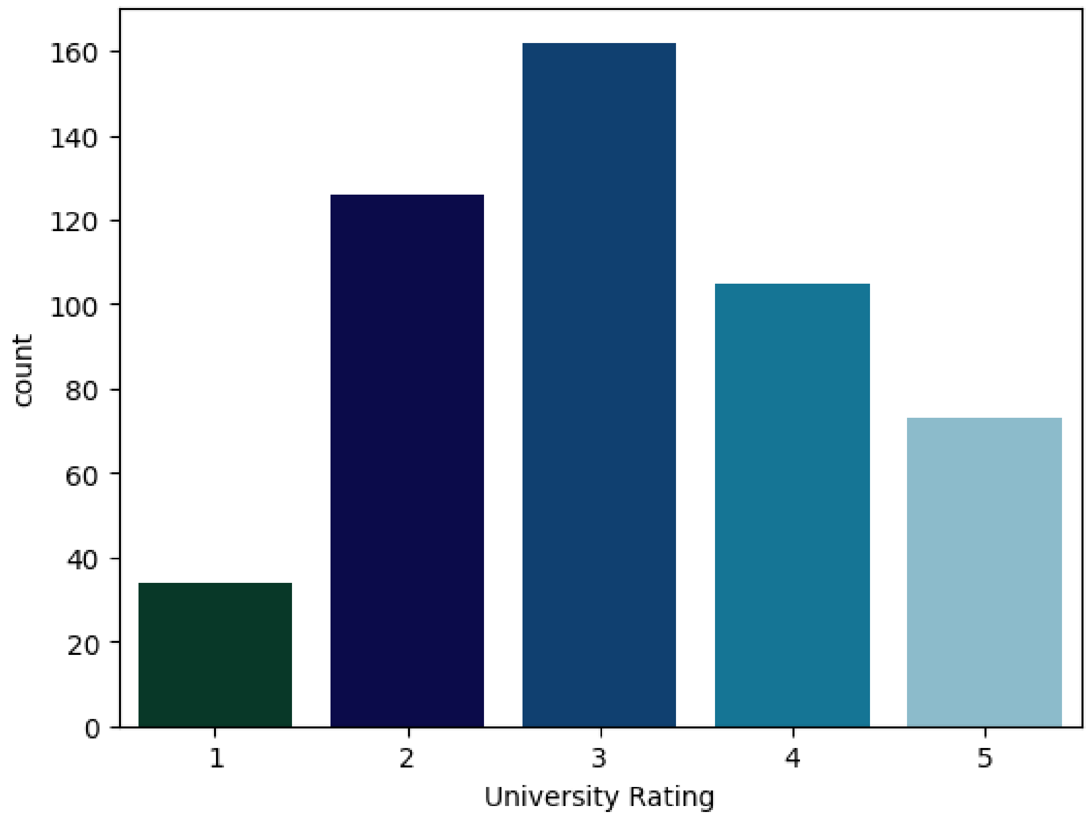

Count
TOEFL Score
0
20
40
60
80
95
100
105
110
115
120

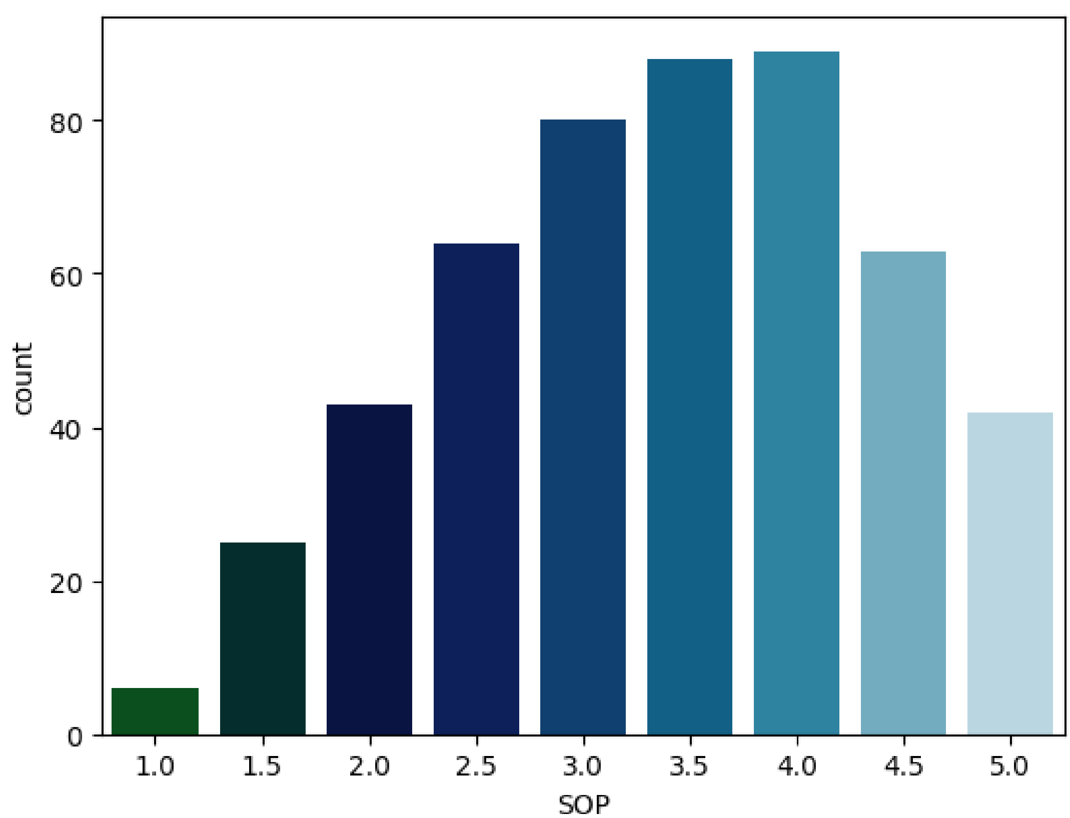

Count
CGPA
0
10
20
30
40
50
60
70
7.0
7.5
8.0
8.5
9.0
9.5
10.0

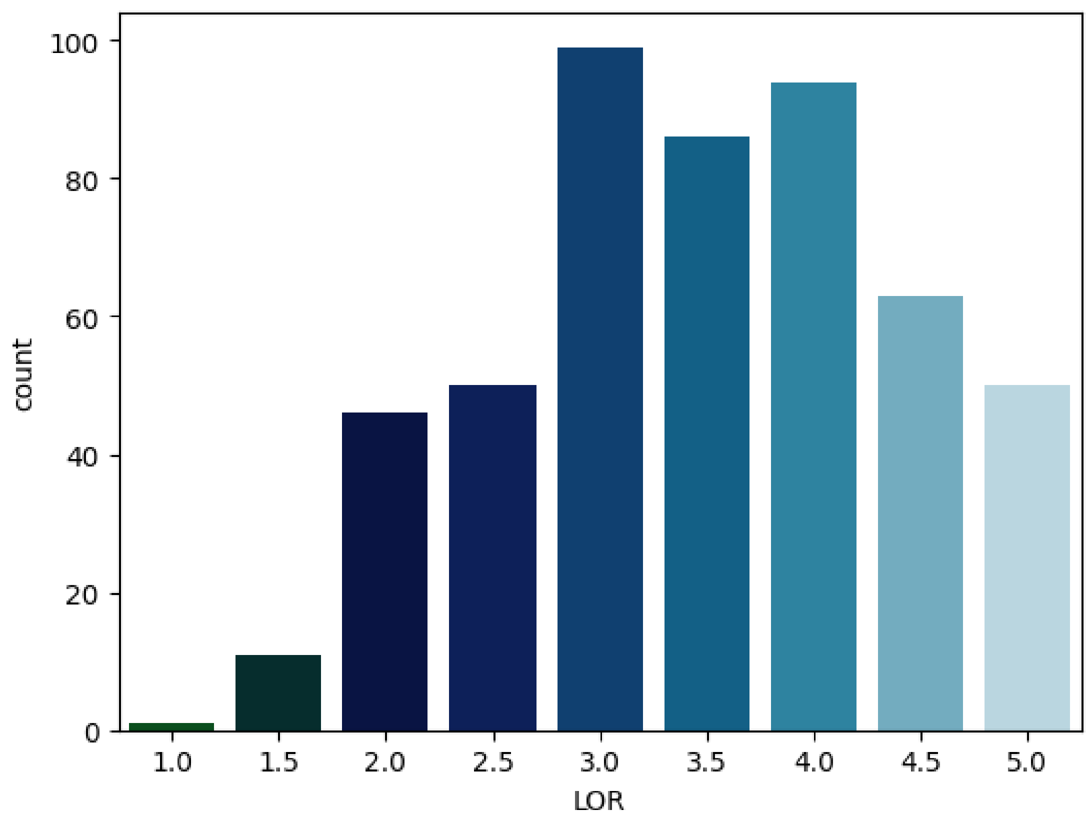


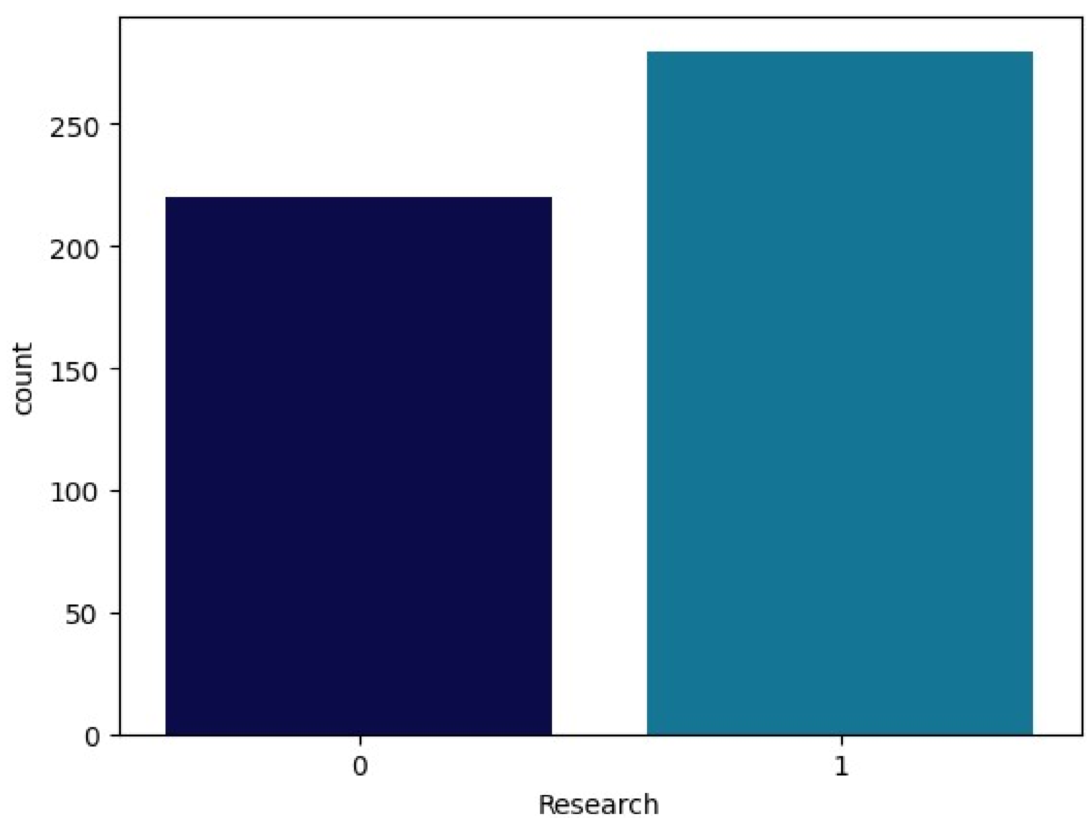
###Categorical Variables

```
for col in categorical_variables:
  sns.countplot(data = data, x = col, palette = 'ocean')
  plt.show()
```


160
140
120
100
80
60
40
20
0
count
1
2
3
4
5
University Rating

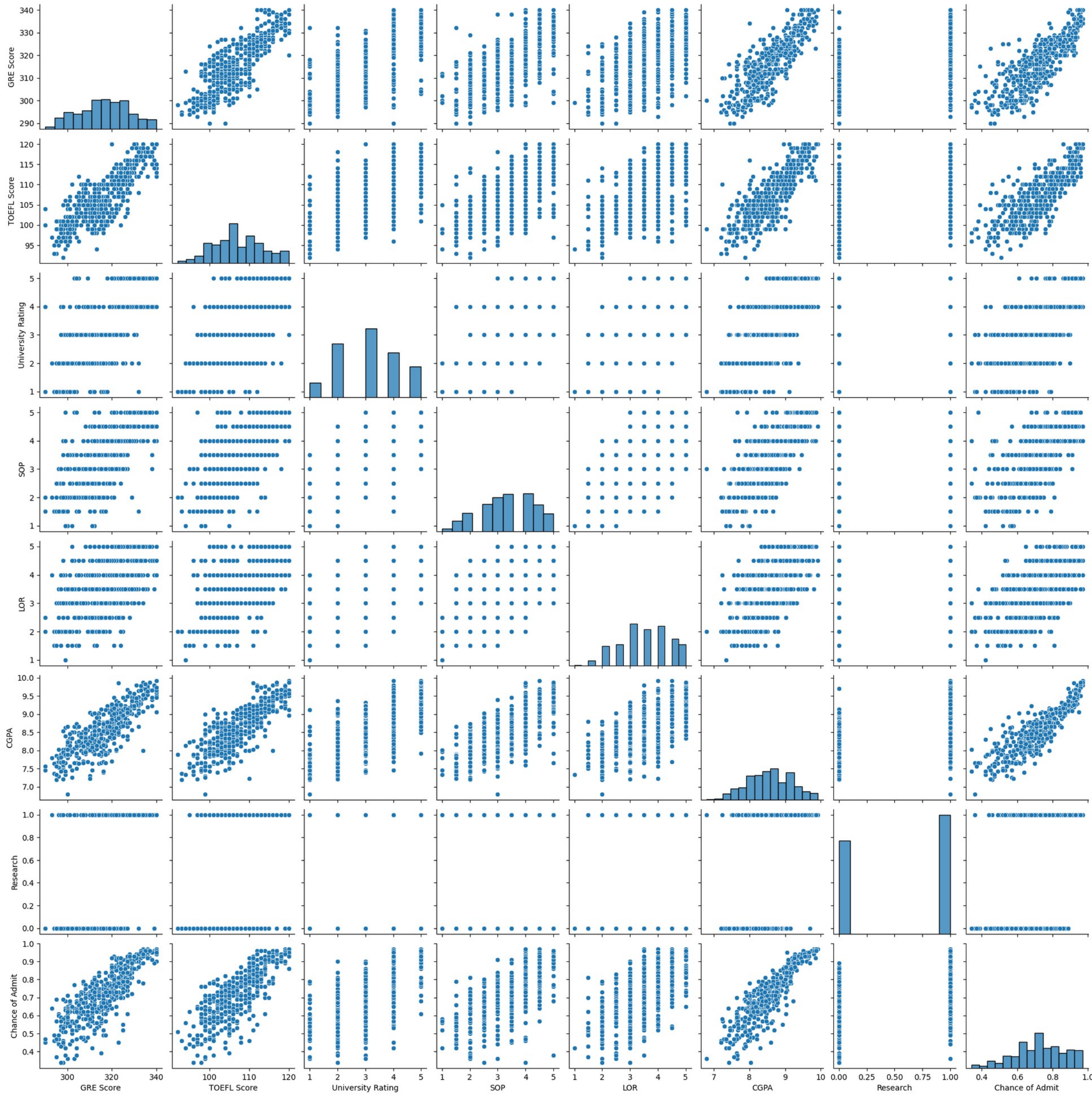

80
60
40
20
0
count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
SOP


count
0
20
40
60
80
100
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
LOR


count
0
50
100
150
200
250
0
1
Research

##Bivariate Analysis

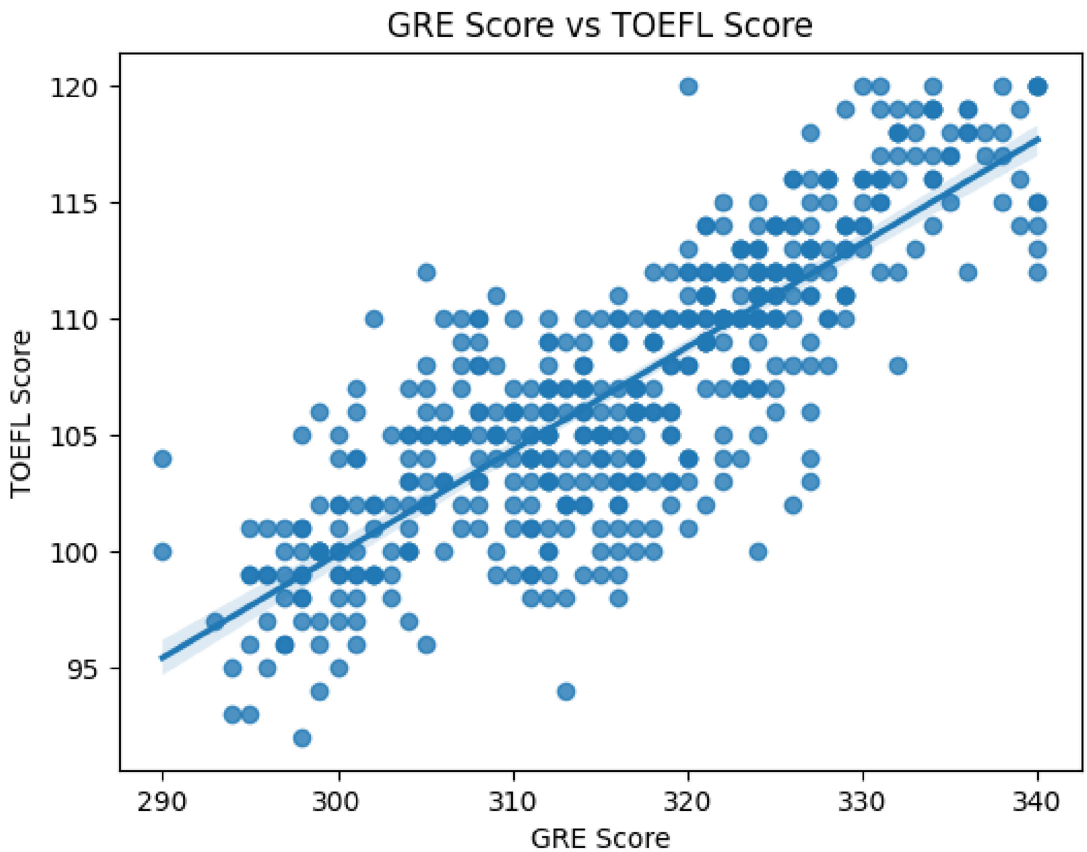
```
sns.pairplot(data)
plt.show()
```



###Relationship among Features

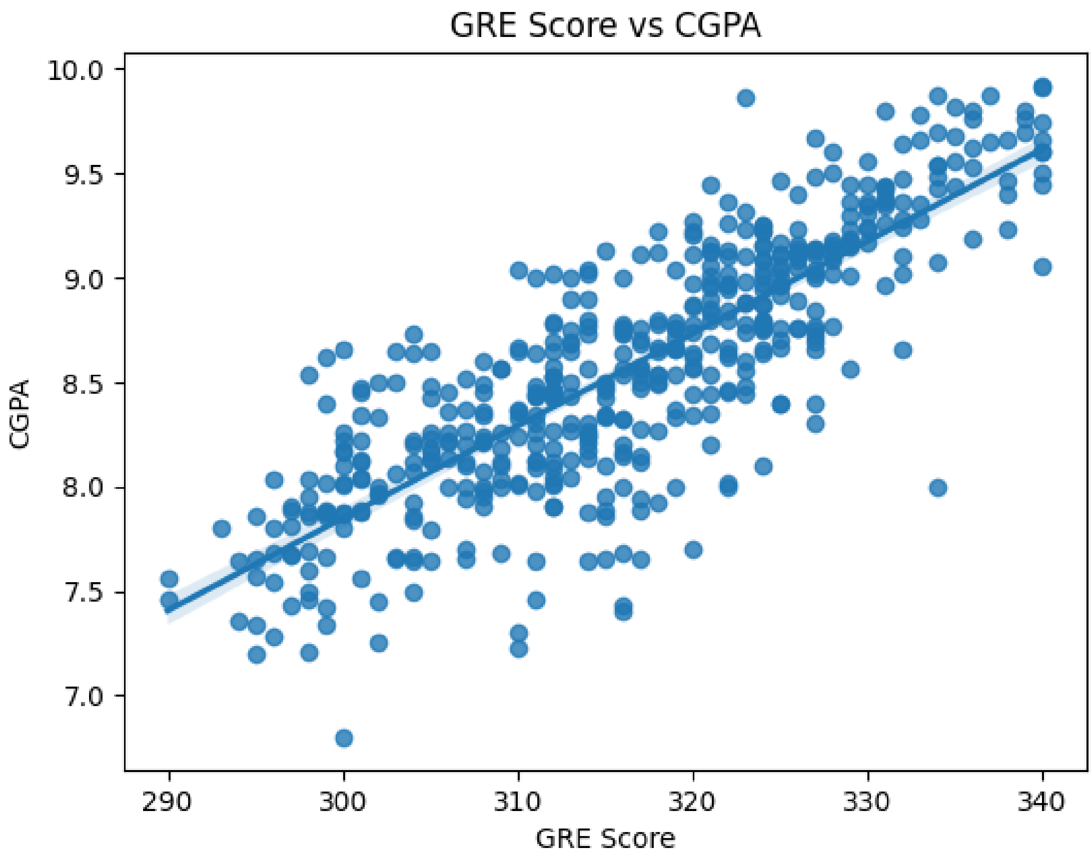
Understanding the relation between different factors responsible for graduate admissions

```
fig = sns.regplot(x="GRE Score", y="TOEFL Score", data=data)
plt.title("GRE Score vs TOEFL Score")
plt.show()
```



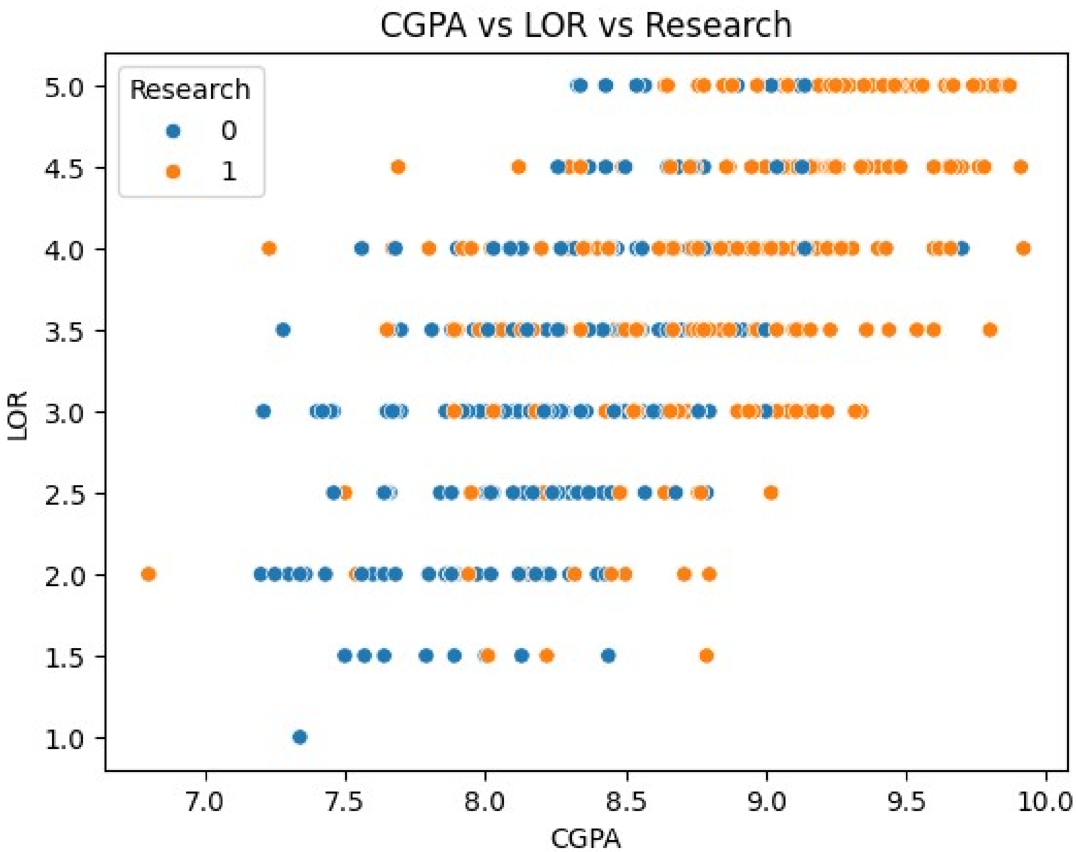
People with higher GRE Scores also have higher TOEFL Scores which is justified because both TOEFL and GRE have a verbal section which although not similar are relatable

```
fig = sns.regplot(x="GRE Score", y="CGPA", data=data)
plt.title("GRE Score vs CGPA")
plt.show()
```



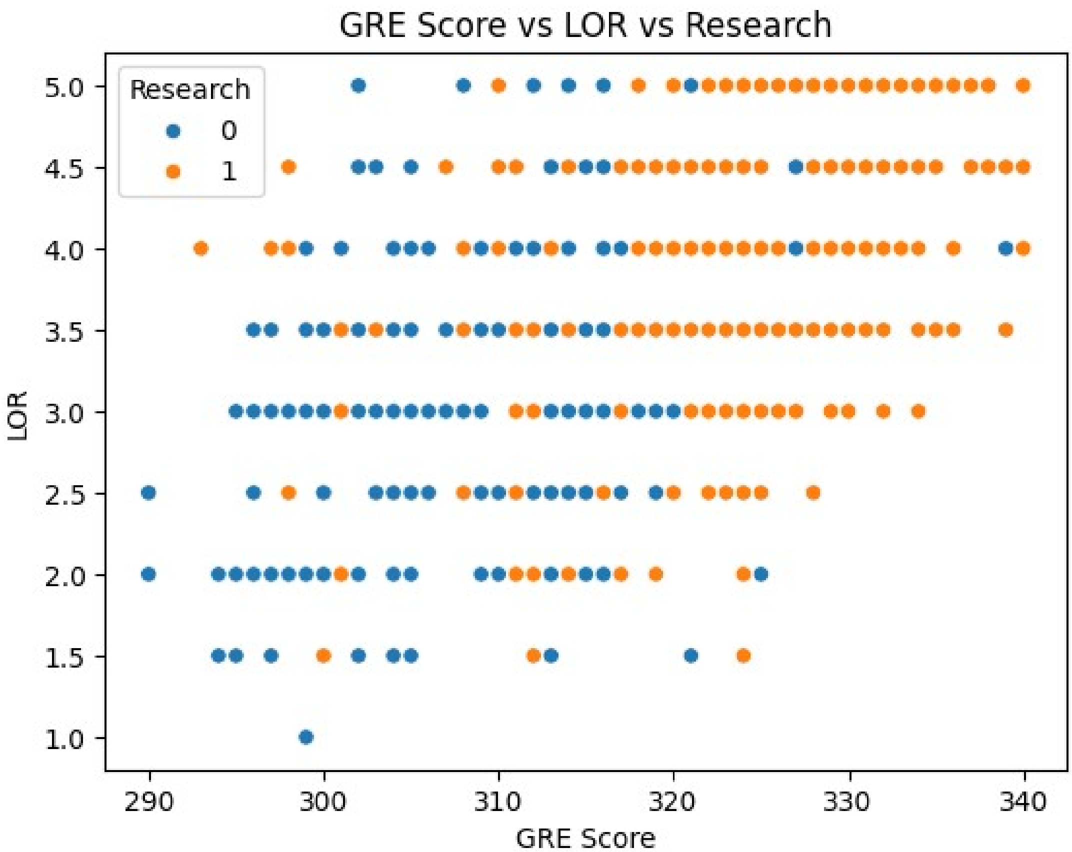
Although there are exceptions, people with higher CGPA usually have higher GRE scores maybe because they are smart or hard working

```
fig = sns.scatterplot(x="CGPA", y="LOR", data=data, hue="Research")
plt.title("CGPA vs LOR vs Research")
plt.show()
```



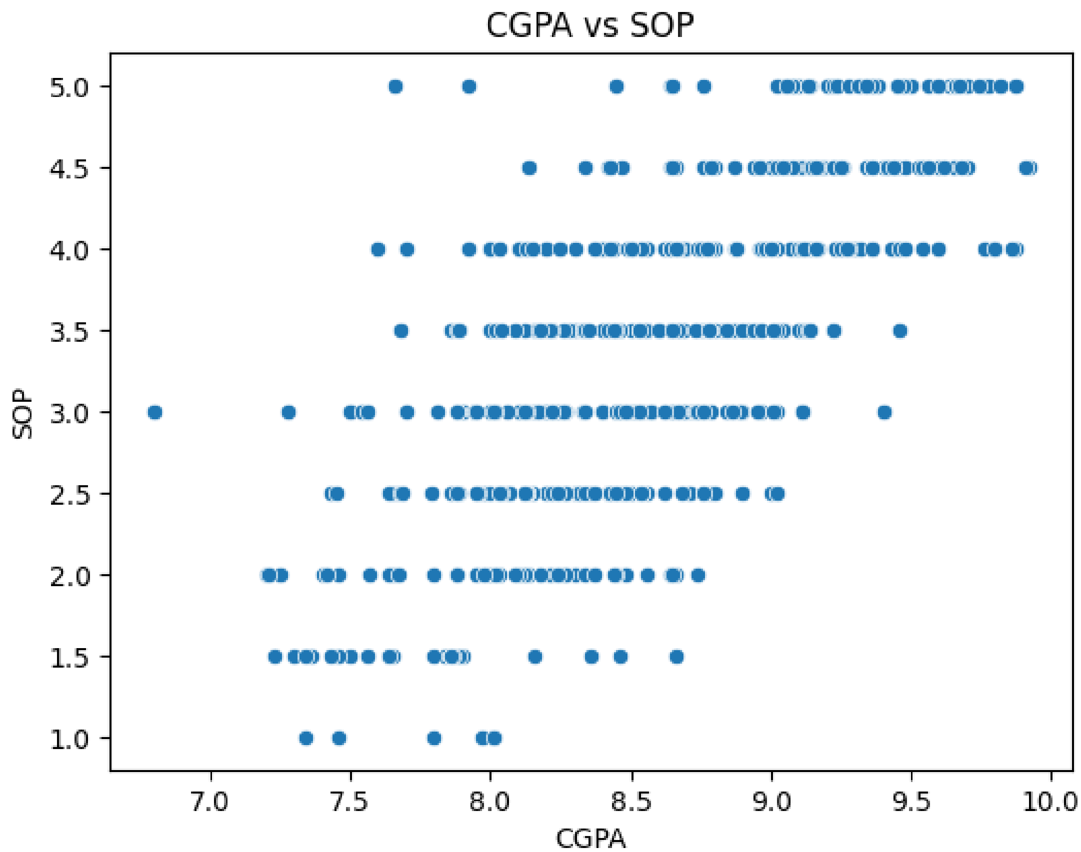
LORs are not that related with CGPA so it is clear that a persons LOR is not dependent on that persons academic excellence. Having research experience is usually related with a good LOR which might be justified by the fact that supervisors have personal interaction with the students performing research which usually results in good LORs

```
fig = sns.scatterplot(x="GRE Score", y="LOR", data=data,
hue="Research")
plt.title("GRE Score vs LOR vs Research")
plt.show()
```



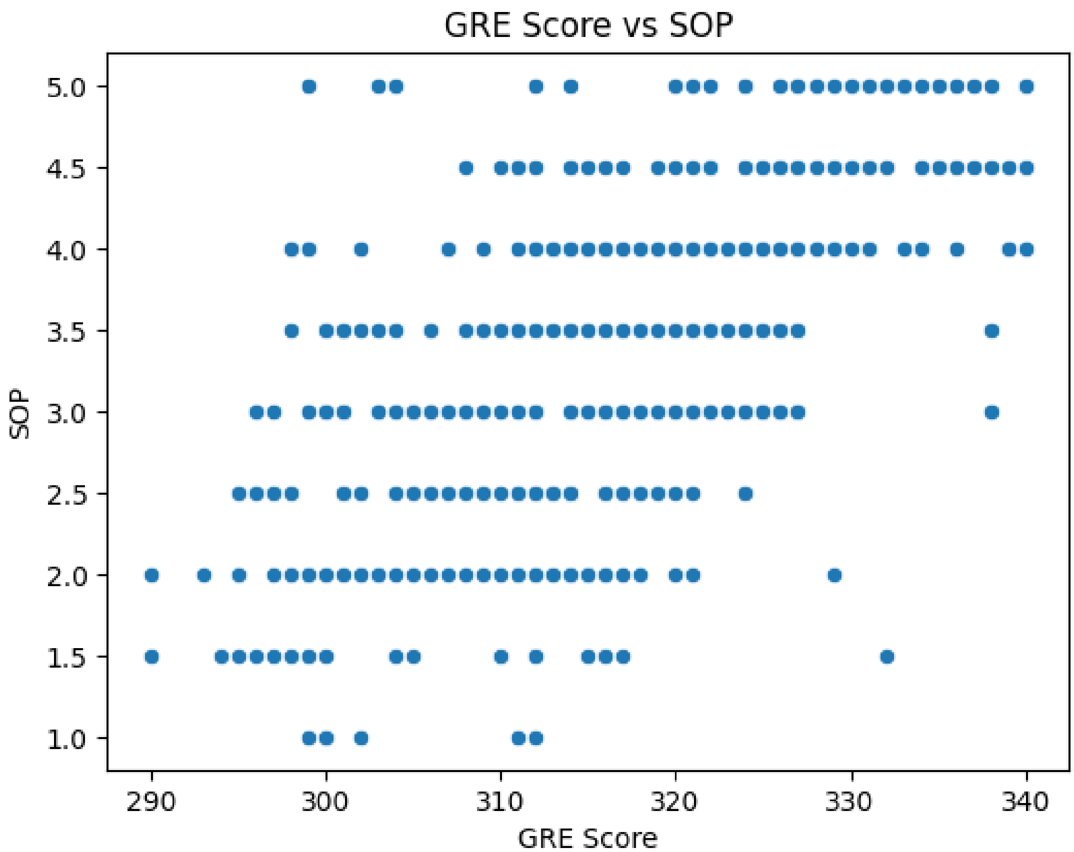
GRE scores and LORs are also not that related. People with different kinds of LORs have all kinds of GRE scores

```
fig = sns.scatterplot(x="CGPA", y="SOP", data=data)
plt.title("CGPA vs SOP")
plt.show()
```



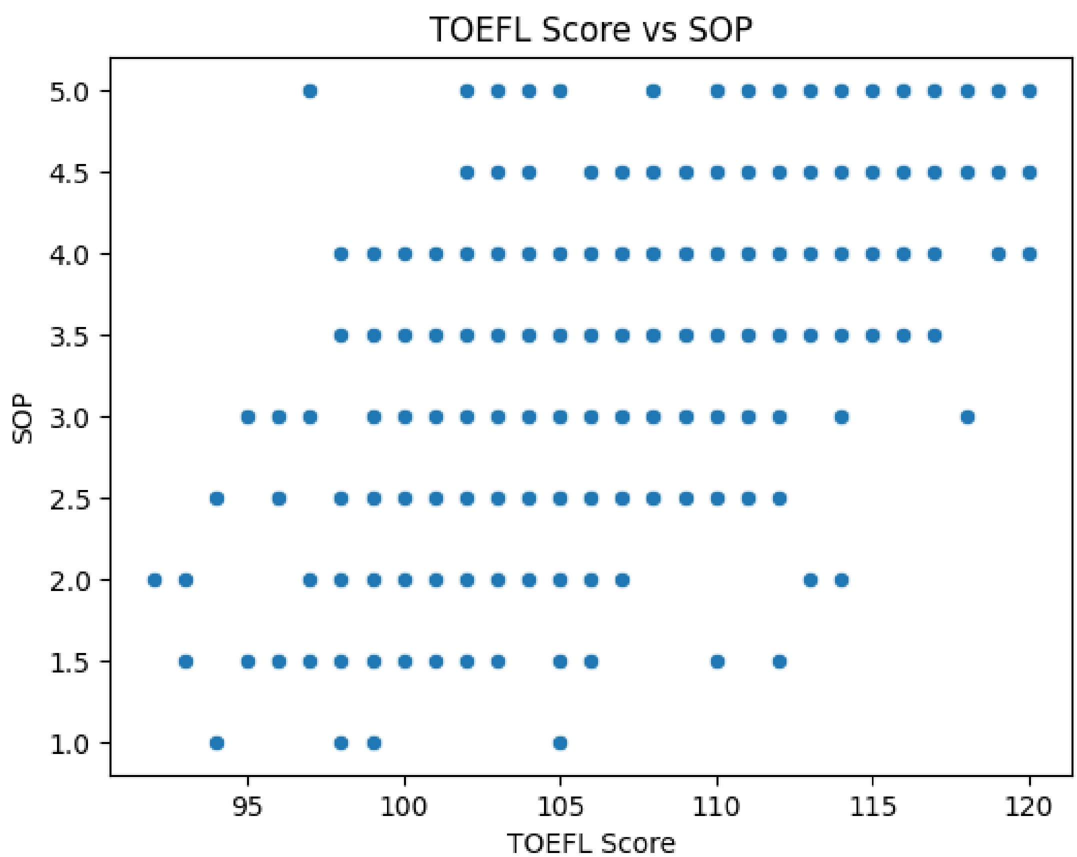
CGPA and SOP are not that related because Statement of Purpose is related to academic performance, but since people with good CGPA tend to be more hard working so they have good things to say in their SOP which might explain the slight move towards higher CGPA as along with good SOPs

```
fig = sns.scatterplot(x="GRE Score", y="SOP", data=data)
plt.title("GRE Score vs SOP")
plt.show()
```



Similary, GRE Score and SOP are only slightly related

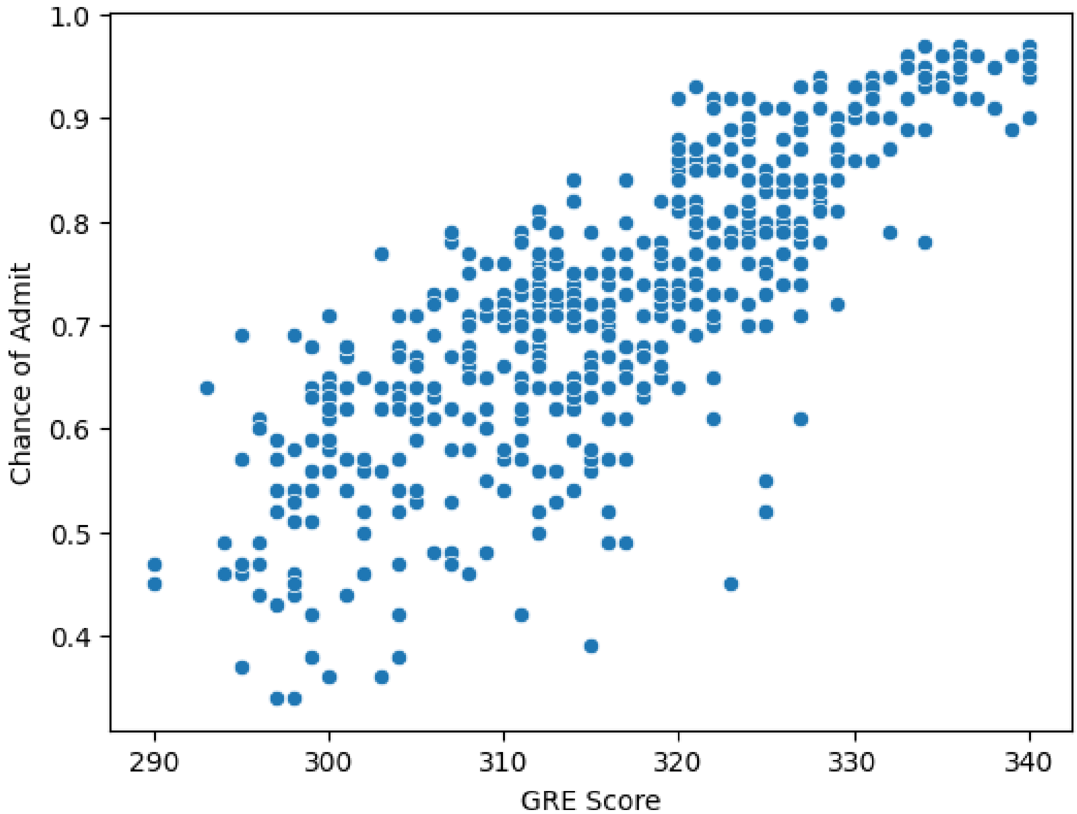
```
fig = sns.scatterplot(x="TOEFL Score", y="SOP", data=data)
plt.title("TOEFL Score vs SOP")
plt.show()
```



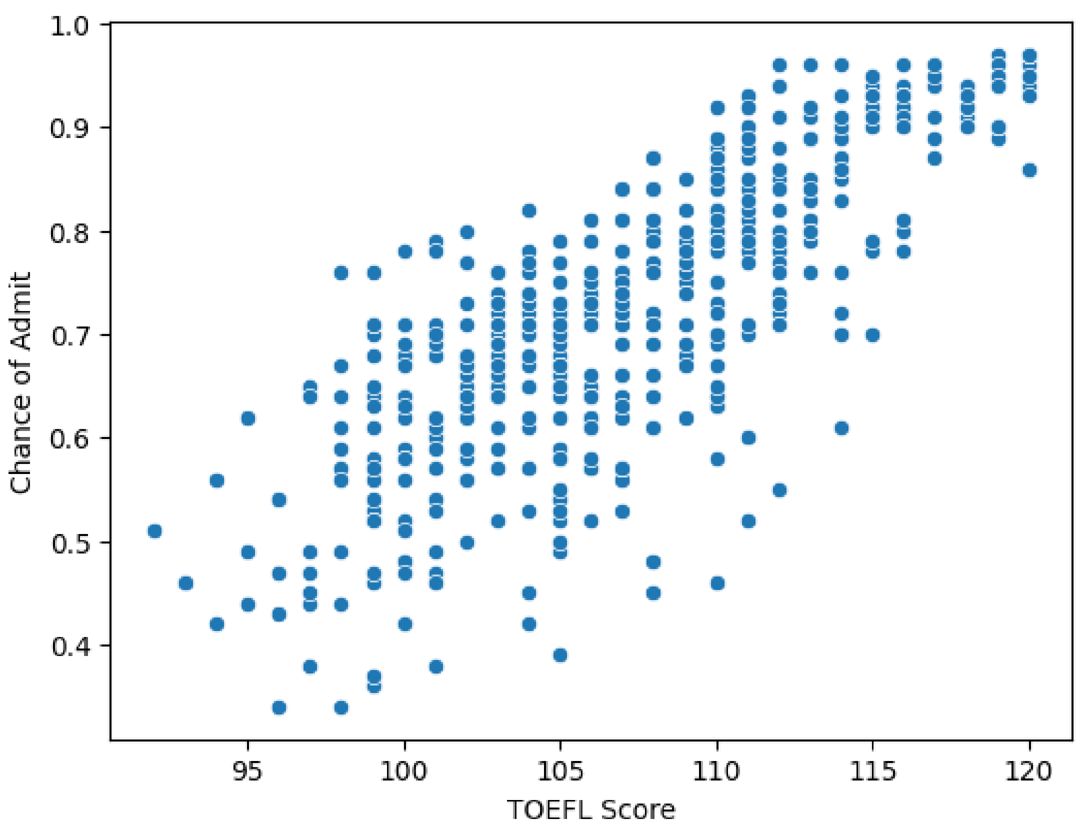
Applicants with different kinds of SOP have different kinds of TOEFL Score. So the quality of SOP is not always related to the applicants English skills.

###Target v/s Continuous Feature (Continuous v/s Continuous)

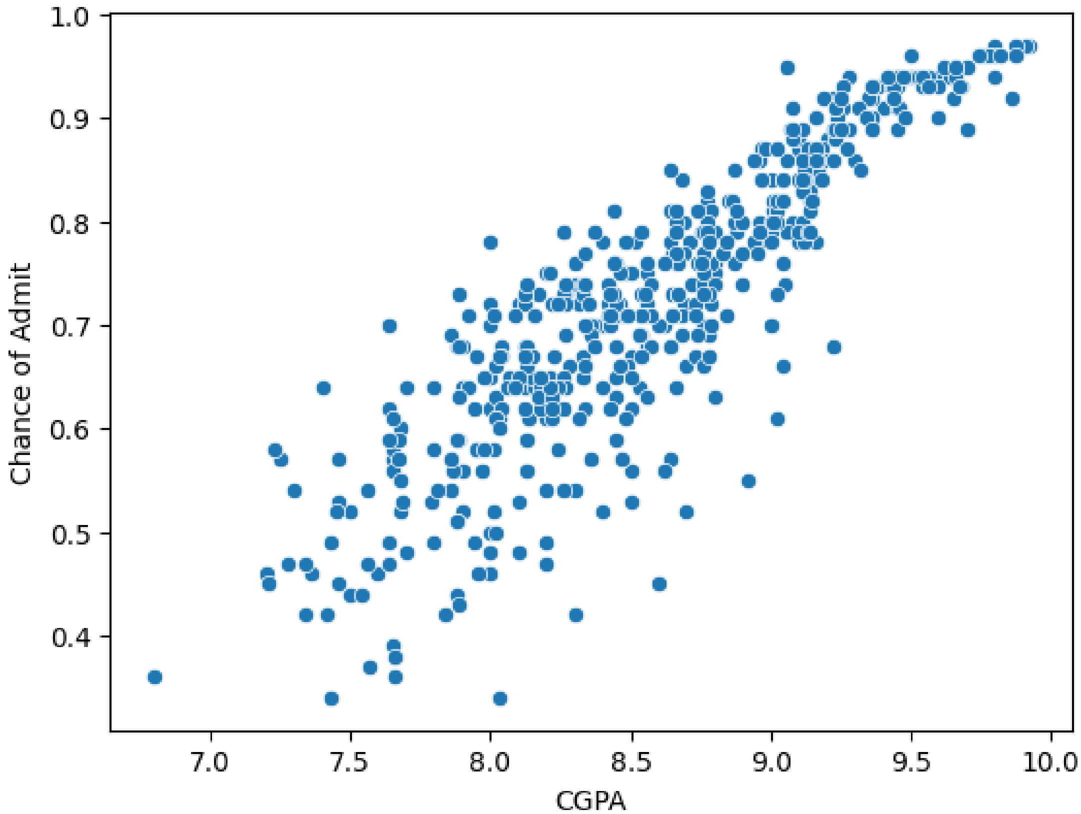
```
for col in data.columns:
  if col not in categorical_variables and col not in ('Chance of
Admit'):
    sns.scatterplot(data = data, x = col, y = 'Chance of Admit')
    plt.show()
```



1.0
0.9
0.8
0.7
0.6
0.5
0.4
Chance of Admit
290
300
310
320
330
340
GRE Score

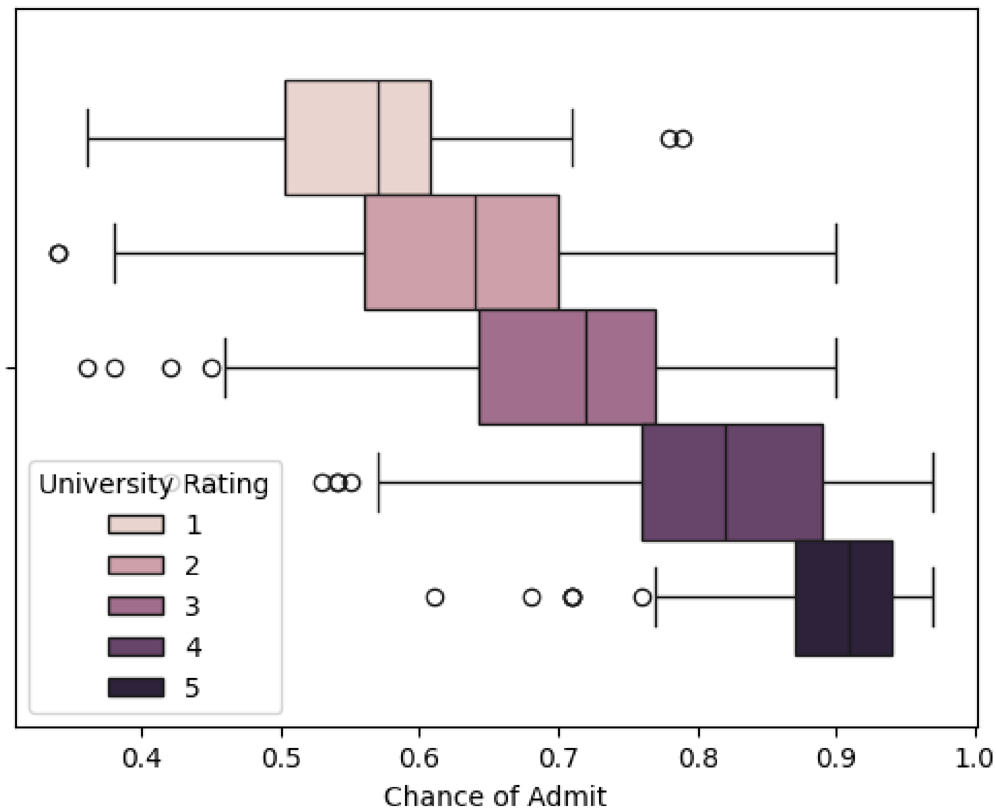


1.0
0.9
0.8
0.7
0.6
0.5
0.4
Chance of Admit
95
100
105
110
115
120
TOEFL Score

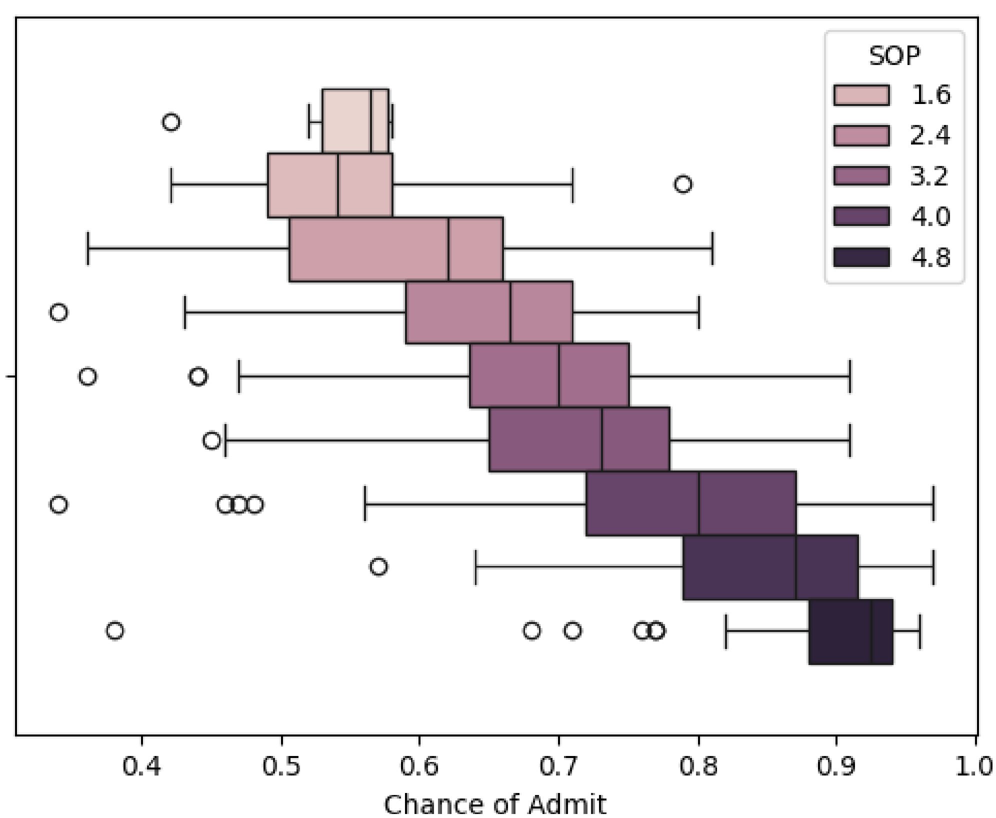


###Target v/s Categorical Features (Continuous v/s Categorical)

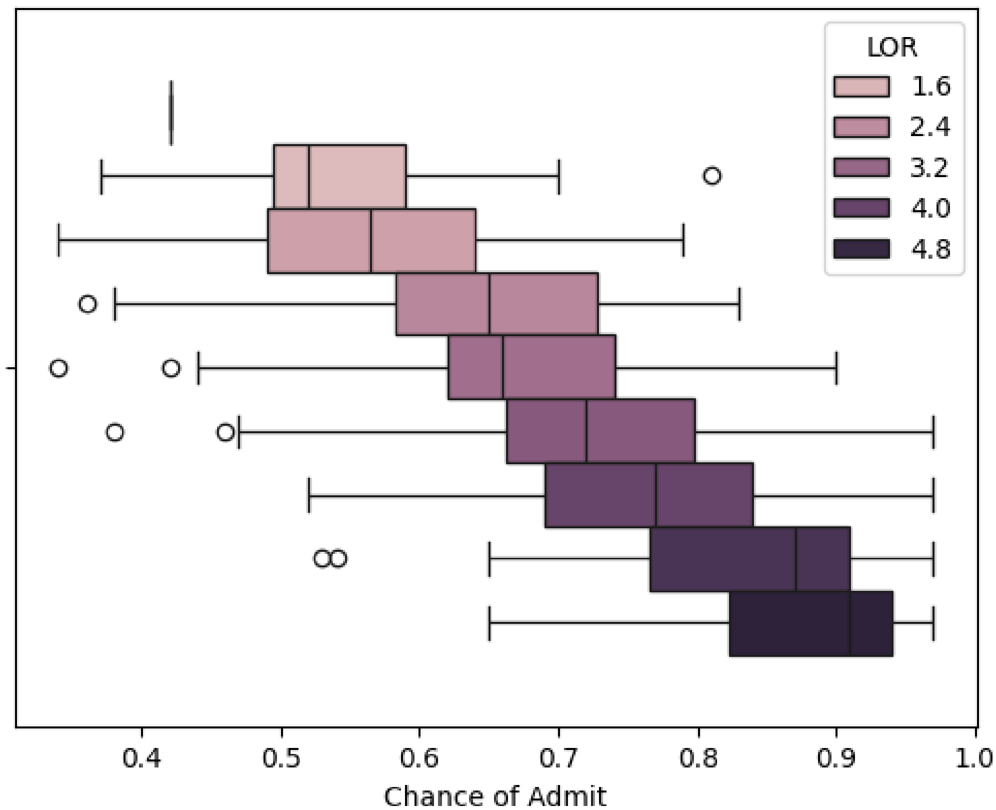
```
for col in categorical_variables:
  sns.boxplot(data = data, x = 'Chance of Admit', hue = col, orient =
'v')
  plt.show()
```



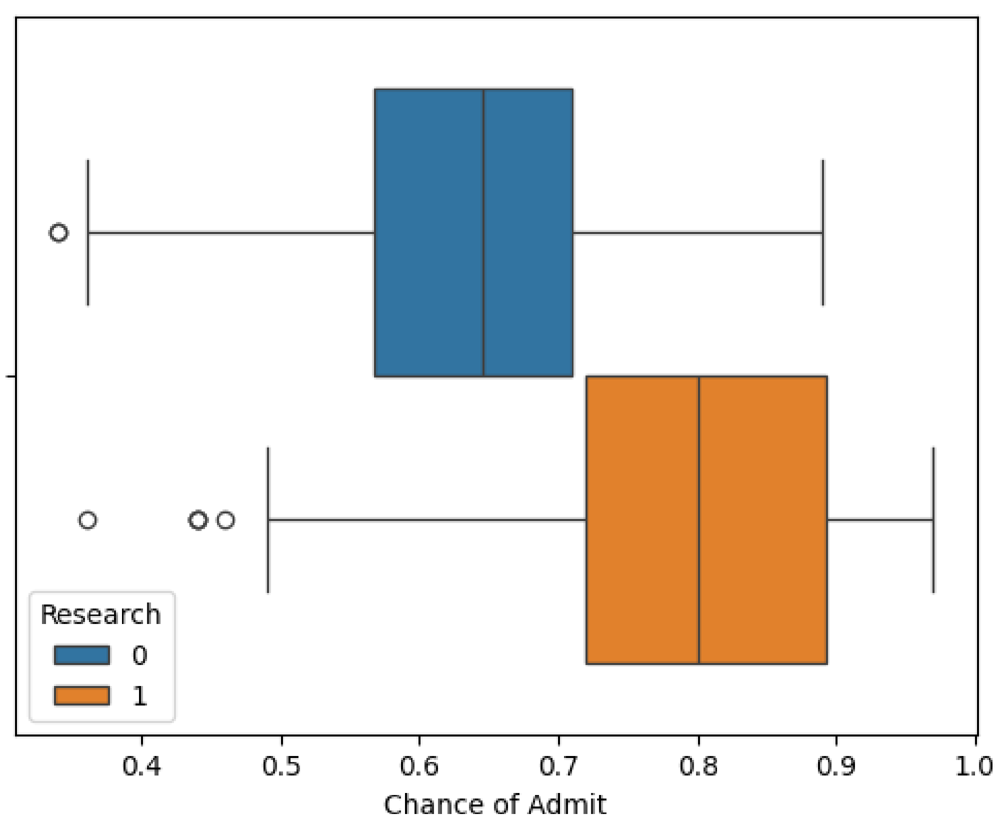
University Rating
1
2
3
4
5
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Chance of Admit



SOP
1.6
2.4
3.2
4.0
4.8
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Chance of Admit



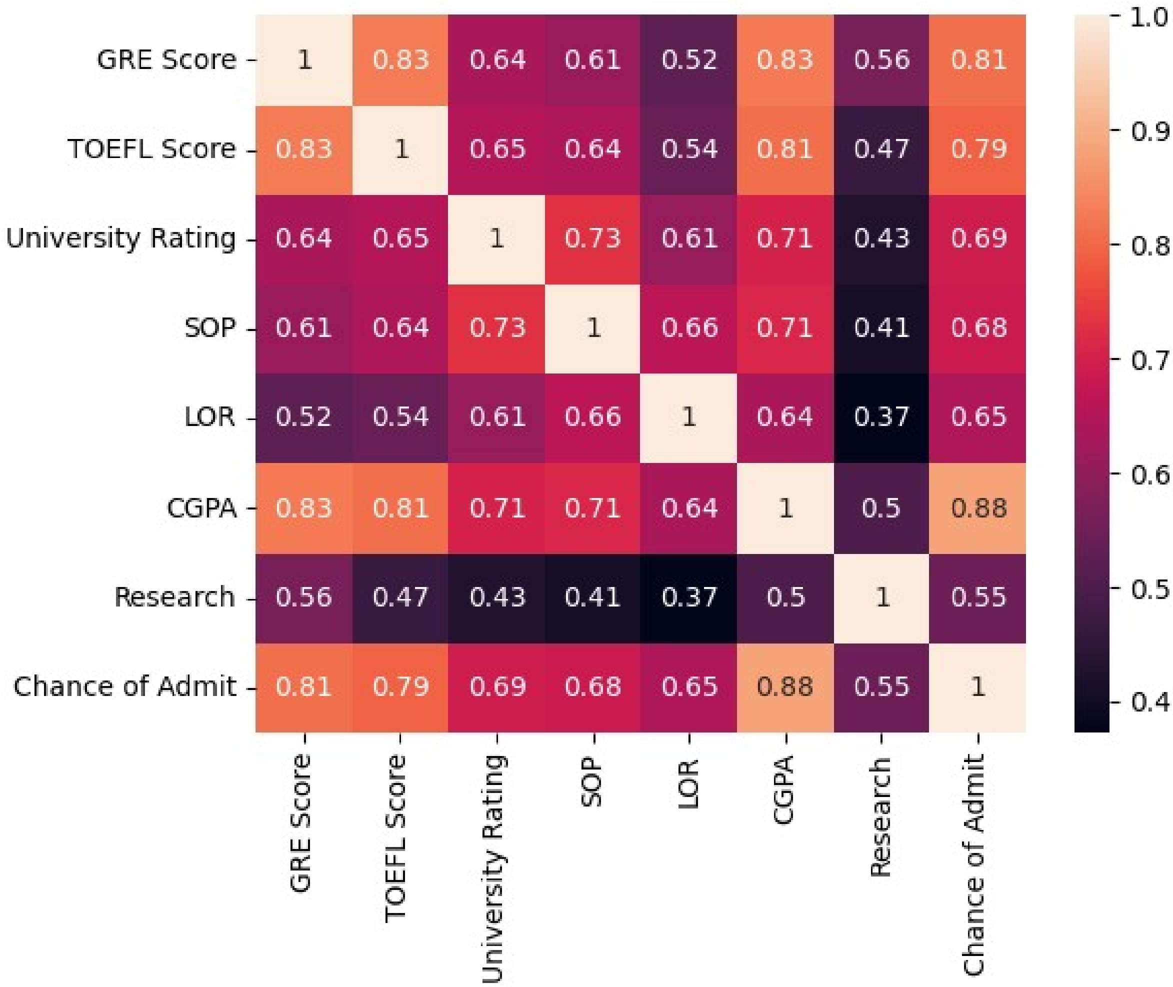
LOR
1.6
2.4
3.2
4.0
4.8
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Chance of Admit



Research
0
1
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Chance of Admit

###Correlation using Heatmap

```
sns.heatmap(data.corr(), annot = True)
plt.show()
```



#Model Building

```
features_data = data.drop(["Chance of Admit"], axis=1)
target_data = data["Chance of Admit"]

features_data.head()
```


{"summary":"{\n \"name\": \"features_data\",\n \"rows\": 500,\n \"fields\": [\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 11,\n   \"min\": 290,\n   \"max\": 340,\n   \"num_unique_values\": 49,\n   \"samples\": [\n   307,\n

335,\n   297\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"TOEFL Score\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 6,\n   \"min\": 92,\n
\"max\": 120,\n   \"num_unique_values\": 29,\n
\"samples\": [\n   94,\n   119,\n   112\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"University Rating\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1,\n   \"min\": 1,\n
\"max\": 5,\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   3,\n   1,\n   2\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.9910036207566072,\n
\"min\": 1.0,\n   \"max\": 5.0,\n   \"num_unique_values\":
9,\n   \"samples\": [\n   1.0,\n   4.0,\n
5.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"LOR\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0.9254495738978193,\n   \"min\": 1.0,\n   \"max\":
5.0,\n   \"num_unique_values\": 9,\n   \"samples\": [\n
5.0,\n   3.5,\n   1.5\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"CGPA\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.6048128003332054,\n
\"min\": 6.8,\n   \"max\": 9.92,\n   \"num_unique_values\":
184,\n   \"samples\": [\n   9.6,\n   8.9,\n
8.24\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Research\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0,\n   \"min\": 0,\n
\"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   0,\n   1\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"features_data"}

```
target_data.head()
```

```
0    0.92
1    0.76
2    0.72
3    0.80
4    0.65
Name: Chance of Admit, dtype: float64
```

```
X_train, X_test, y_train, y_test = train_test_split(features_data,
target_data, test_size = 0.2, shuffle = True)
```

```
X_train.shape
```

(400, 7)

```
y_train.shape
```

(400,)

```
X_test.shape
```

(100, 7)

```
y_test.shape
```

(100,)

```
X_train.head()
```

{"summary":"{\n \"name\": \"X_train\",\n \"rows\": 400,\n
\"fields\": [\n {\n \"column\": \"GRE Score\",\n
\"properties\": {\n \"dtype\": \"number\",\n \"std\":
11,\n \"min\": 290,\n \"max\": 340,\n
\"num_unique_values\": 49,\n \"samples\": [\n 310,\n
295,\n 313\n],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\":
\"TOEFL Score\",\n \"properties\": {\n \"dtype\":
\"number\",\n \"std\": 6,\n \"min\": 92,\n
\"max\": 120,\n \"num_unique_values\": 29,\n
\"samples\": [\n 96,\n 104,\n 110\
n],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\":
\"University Rating\",\n \"properties\": {\n \"dtype\":
\"number\",\n \"std\": 1,\n \"min\": 1,\n
\"max\": 5,\n \"num_unique_values\": 5,\n \"samples\":
[\n 4,\n 1,\n 2\n],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n }\
n },\n {\n \"column\": \"SOP\",\n \"properties\": {\n
\"dtype\": \"number\",\n \"std\": 1.0049220970068435,\n
\"min\": 1.0,\n \"max\": 5.0,\n \"num_unique_values\":
9,\n \"samples\": [\n 4.5,\n 4.0,\n
3.0\n],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\":
\"LOR\",\n \"properties\": {\n \"dtype\": \"number\",\n
\"std\": 0.9208791906158338,\n \"min\": 1.0,\n \"max\":
5.0,\n \"num_unique_values\": 9,\n \"samples\": [\n
1.5,\n 4.0,\n 4.5\n],\n
\"semantic_type\": \"\",\n \"description\": \"\"\n }\
n },\n {\n \"column\": \"CGPA\",\n \"properties\": {\n
\"dtype\": \"number\",\n \"std\": 0.6180260414981918,\n
\"min\": 6.8,\n \"max\": 9.92,\n \"num_unique_values\":
170,\n \"samples\": [\n 7.98,\n 7.92,\n
9.19\n],\n \"semantic_type\": \"\",\n
\"description\": \"\"\n }\n },\n {\n \"column\":

\"Research\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0,\n   \"min\": 0,\n
\"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   0,\n   1\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"X_train"}

```
y_train.head()
```

```
461    0.68
174    0.87
480    0.80
36     0.64
355    0.73
Name: Chance of Admit, dtype: float64
```

```
X_test.head()
```

{"summary":"{\n \"name\": \"X_test\",\n \"rows\": 100,\n
\"fields\": [\n   {\n   \"column\": \"GRE Score\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
9,\n   \"min\": 290,\n   \"max\": 338,\n
\"num_unique_values\": 35,\n   \"samples\": [\n   297,\n
311,\n   307\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"TOEFL Score\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 5,\n   \"min\": 97,\n
\"max\": 120,\n   \"num_unique_values\": 21,\n
\"samples\": [\n   102,\n   120,\n   114\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"University Rating\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0,\n   \"min\": 1,\n   \"max\": 5,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   2,\n
5,\n   1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"SOP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0.922009224520563,\n   \"min\": 1.5,\n   \"max\":
5.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n
3.0,\n   2.5,\n   2.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"LOR\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.924129401931976,\n
\"min\": 1.5,\n   \"max\": 5.0,\n   \"num_unique_values\":
8,\n   \"samples\": [\n   4.5,\n   1.5,\n
3.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"CGPA\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0.5300998952932353,\n   \"min\": 7.28,\n   \"max\":
9.8,\n   \"num_unique_values\": 71,\n   \"samples\": [\n

9.02,\n   8.27,\n   8.8\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Research\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 0,\n
\"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   1,\n   0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"X_test"}

```
y_test.head()
```

```
167    0.64
428    0.69
7      0.68
68     0.68
9      0.45
Name: Chance of Admit, dtype: float64
```

```
scaler = StandardScaler()
```

```
X_train = pd.DataFrame(scaler.fit_transform(X_train), columns =
X_train.columns) #for improving model interpretability
X_train.head()
```

{"summary":"{\n \"name\": \"X_train\",\n \"rows\": 400,\n
\"fields\": [\n   {\n   \"column\": \"GRE Score\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.0012523486435176,\n   \"min\": -2.3289126451775934,\n
\"max\": 1.985886141626654,\n   \"num_unique_values\": 49,\n
\"samples\": [\n   -0.6029931304558945,\n   -
1.8974327664971689,\n   -0.3441052032476397\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"TOEFL Score\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.0012523486435176,\n   \"min\": -2.4702114737980225,\n
\"max\": 1.9928258719896608,\n   \"num_unique_values\": 29,\n
\"samples\": [\n   -1.8326347101140679,\n   -
0.5574811827461583,\n   0.39888396277977384\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"University Rating\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.0012523486435176,\n   \"min\": -1.8617881762660737,\n
\"max\": 1.5504238306362377,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   0.6973708289106598,\n   -
1.8617881762660737,\n   -1.008735174540496\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 1.0012523486435176,\n
\"min\": -2.403690095279134,\n   \"max\": 1.5817028088106218,\n
\"num_unique_values\": 9,\n   \"samples\": [\n

1.0835286957994021,\n   0.5853545827881828,\n   -
0.41099364323425613\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"LOR\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 1.0012523486435176,\n   \"min\": -2.752174262738006,\n
\"max\": 1.5969406215887196,\n   \"num_unique_values\": 9,\n
\"samples\": [\n   -2.208534902197165,\n
0.5096619005070382,\n   1.053301261047879\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"CGPA\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 1.0012523486435176,\n
\"min\": -2.9330759432197167,\n   \"max\": 2.1215772882119794,\n
\"num_unique_values\": 170,\n   \"samples\": [\n   -
1.0213801697936256,\n   -1.1185850396288513,\n
0.9389180385500757\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Research\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1.0012523486435176,\n   \"min\": -
1.2120791238484125,\n   \"max\": 0.8250286473253902,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   -
1.2120791238484125,\n   0.8250286473253902\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"X_train"}

```
y_train = y_train.reset_index(drop = True)
y_train.head()
```

```
0    0.68
1    0.87
2    0.80
3    0.64
4    0.73
Name: Chance of Admit, dtype: float64
```

```
y_test = y_test.reset_index(drop = True)
y_test.head()
```

```
0    0.64
1    0.69
2    0.68
3    0.68
4    0.45
Name: Chance of Admit, dtype: float64
```

Let's use a bunch of different algorithms to see which model performs better

```
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso,Ridge,LinearRegression
from sklearn.metrics import mean_squared_error
```

```
models = [
           ['Linear Regression :', LinearRegression()],

          ['Lasso Regression :', Lasso(alpha=0.1)], #can try with
different alpha values
          ['Ridge Regression :', Ridge(alpha=1.0)] #can try with
different alpha values
          ]

print("Results without removing features with multicollinearity ...")


for name,model in models:
    model.fit(X_train, y_train.values)
    predictions = model.predict(scaler.transform(X_test))
    print(name, (np.sqrt(mean_squared_error(y_test, predictions))))

Results without removing features with multicollinearity ...
Linear Regression : 0.07291444053799043
Lasso Regression : 0.11324145745248729
Ridge Regression : 0.07290273630383767
```

##Linear Regression using Statsmodel library

- Adjusted. R-squared reflects the fit of the model. R-squared values range from 0 to 1, where a higher value generally indicates a better fit, assuming certain conditions are met.
- P >|t| is your p-value. A p-value of less than 0.05 is considered to be statistically significant
- Confidence Interval represents the range in which our coefficients are likely to fall (with a likelihood of 95%)

```
import statsmodels.api as sm
X_train = sm.add_constant(X_train) #Statsmodels default is without
intercept, to add intercept, we need to add constant
model = sm.OLS(y_train.values, X_train).fit()
print(model.summary())

                            OLS Regression Results

======================================================================
========
Dep. Variable:                      y   R-squared:
0.848
Model:                            OLS   Adj. R-squared:
0.846
Method:                 Least Squares   F-statistic:
312.9
Date:                Wed, 08 May 2024   Prob (F-statistic):
```

```
3.65e-156
Time:                        04:12:47   Log-Likelihood:
586.70
No. Observations:                 400   AIC:
-1157.
Df Residuals:                     392   BIC:
-1125.
Df Model:                           7

Covariance Type:            nonrobust

======================================================================
===============
                        coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
---------------
const                 0.7310      0.003    259.295      0.000
0.725       0.737
GRE Score             0.0224      0.006      3.592      0.000
0.010       0.035
TOEFL Score           0.0221      0.006      3.791      0.000
0.011       0.033
University Rating     0.0055      0.005      1.198      0.232      -
0.003       0.014
SOP                   0.0010      0.005      0.201      0.841      -
0.009       0.011
LOR                   0.0137      0.004      3.402      0.001
0.006       0.022
CGPA                  0.0731      0.006     11.853      0.000
0.061       0.085
Research              0.0108      0.004      3.069      0.002
0.004       0.018
======================================================================
========
Omnibus:                       67.892   Durbin-Watson:
2.126
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
130.861
Skew:                          -0.936   Prob(JB):
3.84e-29
Kurtosis:                       5.086   Cond. No.
5.88
======================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

P-value for "SOP" seems to be highest and >0.05 so let's drop that

```
X_train_new=X_train.drop(columns='SOP')

model1 = sm.OLS(y_train.values, X_train_new).fit()
print(model1.summary())
```

```
                            OLS Regression Results                            
==============================================================================
========
Dep. Variable:                      y   R-squared:
0.848
Model:                            OLS   Adj. R-squared:
0.846
Method:                 Least Squares   F-statistic:
366.0
Date:                Wed, 08 May 2024   Prob (F-statistic):
1.86e-157
Time:                        04:14:41   Log-Likelihood:
586.67
No. Observations:                 400   AIC:
-1159.
Df Residuals:                     393   BIC:
-1131.
Df Model:                           6

Covariance Type:            nonrobust

==============================================================================
===============
                        coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
---------------
const                 0.7310      0.003    259.612      0.000
0.725       0.737
GRE Score             0.0224      0.006      3.594      0.000
0.010       0.035
TOEFL Score           0.0222      0.006      3.828      0.000
0.011       0.034
University Rating     0.0058      0.004      1.380      0.168      -
0.002       0.014
LOR                   0.0139      0.004      3.632      0.000
0.006       0.021
CGPA                  0.0733      0.006     12.133      0.000
0.061       0.085
Research              0.0108      0.003      3.078      0.002
0.004       0.018
==============================================================================
```

```
========
Omnibus:                       66.995   Durbin-Watson:
2.126
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
128.348
Skew:                          -0.927   Prob(JB):
1.35e-28
Kurtosis:                       5.065   Cond. No.
5.43
======================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

#Testing the assumptions of the linear regression model

##No Multicollinearity

VIF(Variance Inflation Factor)

- “ VIF score of an independent variable represents how well the variable is explained by other independent variables.
- So, the closer the R^2 value to 1, the higher the value of VIF and the higher the multicollinearity with the particular independent variable.

```
from statsmodels.stats.outliers_influence import
variance_inflation_factor

def calculate_vif(dataset,col):
  dataset=dataset.drop(columns=col,axis=1)
  vif=pd.DataFrame()
  vif['features']=dataset.columns
  vif['VIF_Value']=[variance_inflation_factor(dataset.values,i) for i
in range(dataset.shape[1])]
  return vif

calculate_vif(X_train_new,[])
```


```
{"summary":"{\n  \"name\": \"calculate_vif(X_train_new,[])\",\n
\"rows\": 7,\n  \"fields\": [\n    {\n      \"column\": \"features\",\
n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n
\"const\",\n          \"GRE Score\",\n          \"CGPA\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"VIF_Value\",\n
```

\"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1.6136683063814838,\n   \"min\": 0.9999999999999998,\n   \"max\": 4.891461274378504,\n   \"num_unique_values\": 7,\n   \"samples\": [\n   0.9999999999999998,\n   4.891461274378504,\n   4.6023244225134405\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

VIF looks fine and hence, we can go ahead with the predictions

```
X_test_std= scaler.transform(X_test)

X_test=pd.DataFrame(X_test_std, columns=X_test.columns) # col name same as train datasets

X_test.head()
```

{"summary":"{\n \"name\": \"X_test\",\n \"rows\": 100,\n \"fields\": [\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8416987264960586,\n   \"min\": -2.3289126451775934,\n   \"max\": 1.8132941901544841,\n   \"num_unique_values\": 35,\n   \"samples\": [\n   -1.724840815024999,\n   -0.5166971547198096,\n   -0.8618810576641495\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"TOEFL Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8052853147769465,\n   \"min\": -1.6732405191930793,\n   \"max\": 1.9928258719896608,\n   \"num_unique_values\": 21,\n   \"samples\": [\n   -0.8762695645881357,\n   1.9928258719896608,\n   1.0364607264637287\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"University Rating\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8287524648742667,\n   \"min\": -1.8617881762660737,\n   \"max\": 1.5504238306362377,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   -1.008735174540496,\n   1.5504238306362377,\n   -1.8617881762660737\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.9186422552273876,\n   \"min\": -1.9055159822679144,\n   \"max\": 1.5817028088106218,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   -0.4109936432342561З,\n   -0.9091677562454756,\n   -1.407341869256695\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"LOR\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1.0047862342465779,\n   \"min\": -2.208534902197165,\n   \"max\": 1.5969406215887196,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   1.053301261047879,\n   -

2.208534902197165,\n   -0.5776168205746433\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"CGPA\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8588048553607549,\n   \"min\": -2.1554369845379164,\n   \"max\": 1.927167548541531,\n   \"num_unique_values\": 71,\n   \"samples\": [\n   0.6635042406836051,\n   -0.5515566322567064,\n   0.30708638462111565\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Research\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1.0104970505044262,\n   \"min\": -1.2120791238484125,\n   \"max\": 0.8250286473253902,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0.8250286473253902,\n   -1.2120791238484125\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"X_test"}

```
X_test = sm.add_constant(X_test)

X_test.head()
```

{"summary":"{\n \"name\": \"X_test\",\n \"rows\": 100,\n \"fields\": [\n   {\n   \"column\": \"const\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.0,\n   \"min\": 1.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   1.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8416987264960586,\n   \"min\": -2.3289126451775934,\n   \"max\": 1.8132941901544841,\n   \"num_unique_values\": 35,\n   \"samples\": [\n   -1.724840815024999\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"TOEFL Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8052853147769465,\n   \"min\": -1.6732405191930793,\n   \"max\": 1.9928258719896608,\n   \"num_unique_values\": 21,\n   \"samples\": [\n   -0.8762695645881357\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"University Rating\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8287524648742667,\n   \"min\": -1.8617881762660737,\n   \"max\": 1.5504238306362377,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   -1.008735174540496\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"SOP\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.9186422552273876,\n   \"min\": -1.9055159822679144,\n   \"max\": 1.5817028088106218,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   -0.41099364323425613\n   ],\n   \"semantic_type\": \"\",\n

\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"LOR\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1.0047862342465779,\n   \"min\": -2.208534902197165,\n   \"max\": 1.5969406215887196,\n   \"num_unique_values\": 8,\n   \"samples\": [\n   1.053301261047879\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"CGPA\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8588048553607549,\n   \"min\": -2.1554369845379164,\n   \"max\": 1.927167548541531,\n   \"num_unique_values\": 71,\n   \"samples\": [\n   0.6635042406836051\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Research\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1.0104970505044262,\n   \"min\": -1.2120791238484125,\n   \"max\": 0.8250286473253902,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   0.8250286473253902\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"X_test"}

```
X_test_del=list(set(X_test.columns).difference(set(X_train_new.columns)))
X_test_del
```

```
['SOP']
```

```
print(f'Dropping {X_test_del} from test set')
```

```
Dropping ['SOP'] from test set
```

```
X_test_new=X_test.drop(columns=X_test_del)
```

```
X_test_new.head()
```

{"summary":"{\n \"name\": \"X_test_new\",\n \"rows\": 100,\n \"fields\": [\n   {\n   \"column\": \"const\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.0,\n   \"min\": 1.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   1.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"GRE Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8416987264960586,\n   \"min\": -2.3289126451775934,\n   \"max\": 1.8132941901544841,\n   \"num_unique_values\": 35,\n   \"samples\": [\n   -1.724840815024999\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"TOEFL Score\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8052853147769465,\n   \"min\": -1.6732405191930793,\n   \"max\": 1.9928258719896608,\n   \"num_unique_values\": 21,\n   \"samples\": [\n   -0.8762695645881357\n   ],\n

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"University Rating\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.8287524648742667,\n        \"min\": -1.8617881762660737,\n
\"max\": 1.5504238306362377,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          -1.008735174540496\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"LOR\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1.0047862342465779,\n
\"min\": -2.208534902197165,\n        \"max\": 1.5969406215887196,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
1.053301261047879\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"CGPA\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.8588048553607549,\n        \"min\": -2.1554369845379164,\n
\"max\": 1.927167548541531,\n        \"num_unique_values\": 71,\n
\"samples\": [\n          0.6635042406836051\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Research\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1.0104970505044262,\n        \"min\": -1.2120791238484125,\n
\"max\": 0.8250286473253902,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0.8250286473253902\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"X_test_new"}
```



```
#Prediction from the clean model
pred = model1.predict(X_test_new)

from sklearn.metrics import
mean_squared_error,r2_score,mean_absolute_error

print('Mean Absolute Error ',
mean_absolute_error(y_test.values,pred) )
print('Root Mean Square Error ',
np.sqrt(mean_squared_error(y_test.values,pred) ))
```

```
Mean Absolute Error  0.04822323585173553
Root Mean Square Error  0.07293057291936322
```

```
def r2_score(y, y_pred):
  num = np.sum((y-y_pred)**2)
  denom = np.sum((y-y.mean())**2)
  score = 1 - (num/denom)
  return score

def adj_r2(r2, X, y):
  adj_r_squared = 1 - (((1-r2)*(len(y)-1))/(len(y) - X.shape[1]-1))
  return adj_r_squared
```

```
print("R2 score ", r2_score(y_test, pred))
print("Adj R2 score ", adj_r2(r2_score(y_test, pred), X_test_new,
y_test))
```

```
R2 score  0.6587167672866537
Adj R2 score  0.6327495647975948
```

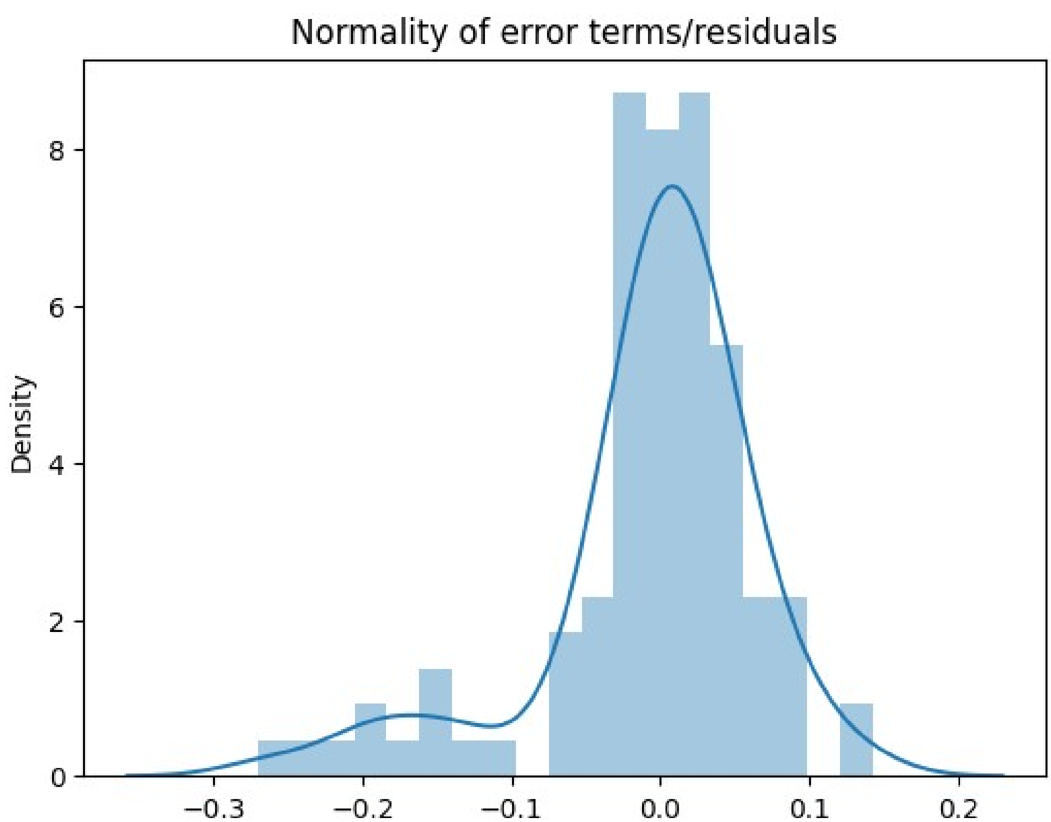
##Mean of Residuals

```
residuals = y_test.values-pred
mean_residuals = np.mean(residuals)
print("Mean of Residuals {}".format(mean_residuals))
```

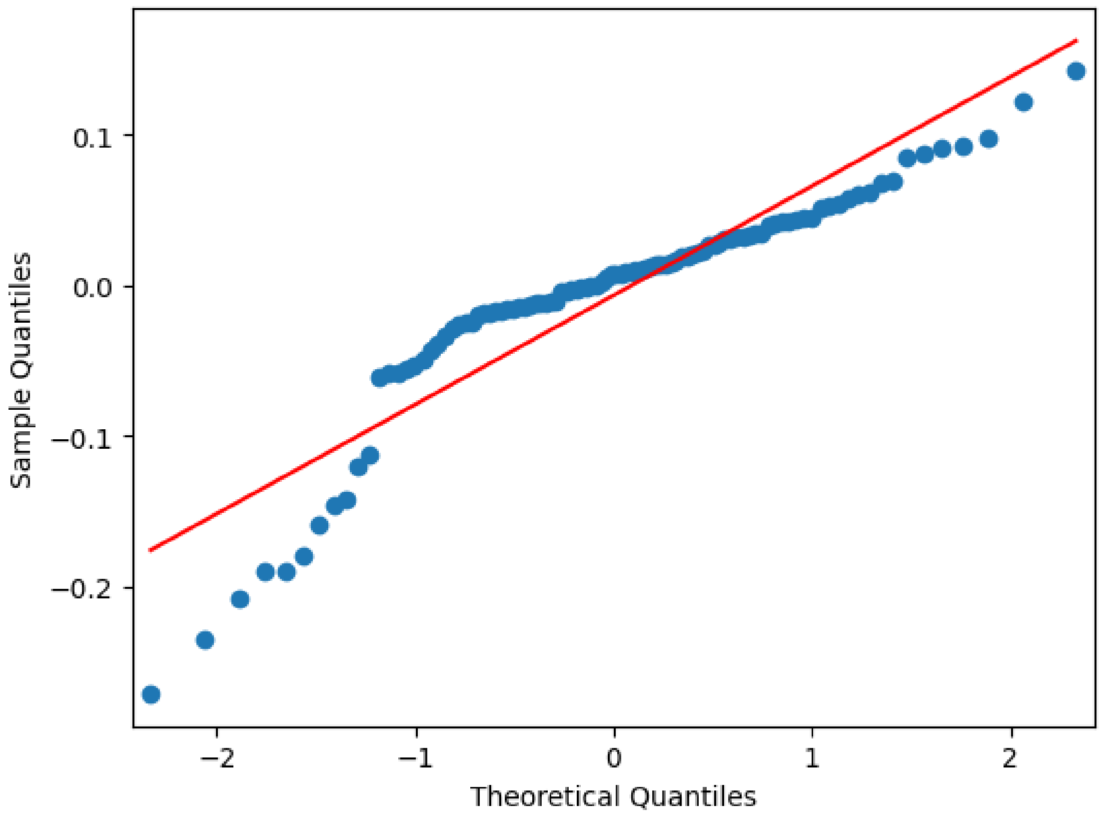
```
Mean of Residuals -0.006628840611761088
```

##Normality of Residuals

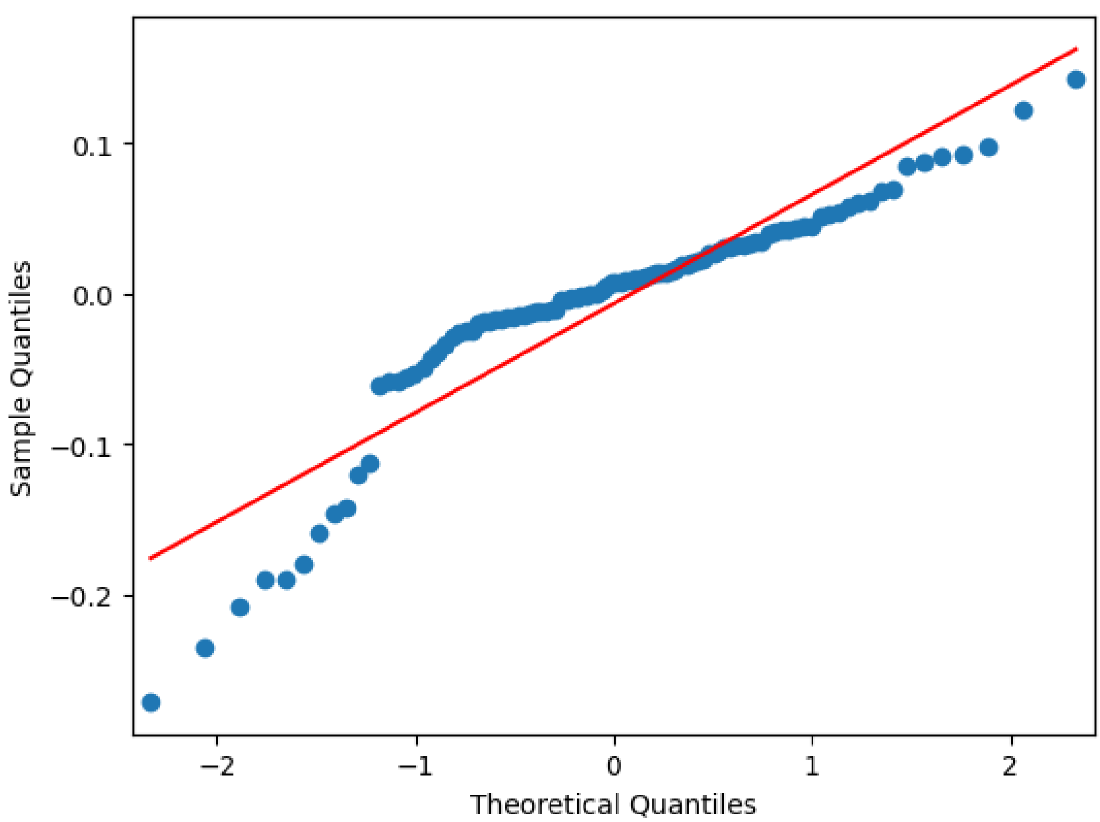
```
p = sns.distplot(residuals,kde=True)
p = plt.title('Normality of error terms/residuals')
```




```
qqplot(residuals, line = "s")
```



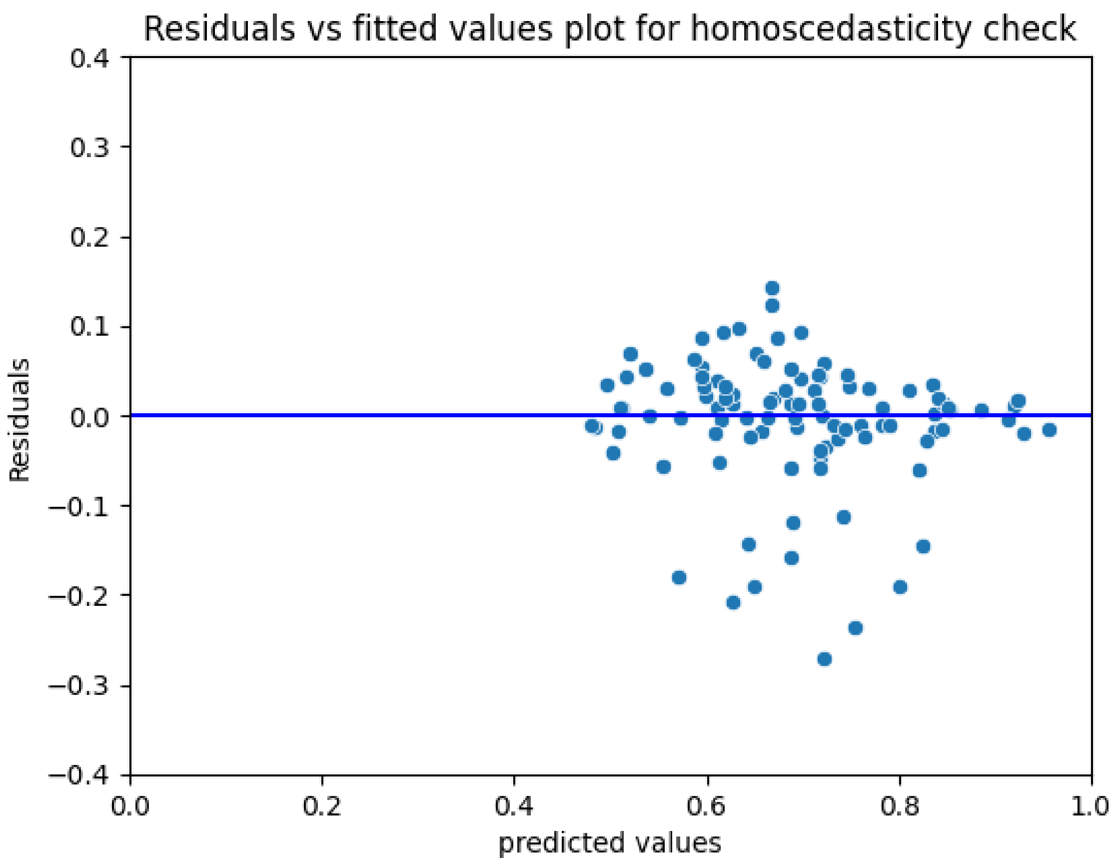
0.1
0.0
−0.1
−0.2
Sample Quantiles
−2
−1
0
1
2
Theoretical Quantiles



0.1
0.0
−0.1
−0.2
Sample Quantiles
−2
−1
0
1
2
Theoretical Quantiles

##Test for Homoscedasticity

```
p = sns.scatterplot(x=pred,y=residuals)
plt.xlabel('predicted values')
plt.ylabel('Residuals')
plt.ylim(-0.4,0.4)
plt.xlim(0,1)
p = sns.lineplot(x=[0,26], y=[0,0], color='blue')
p = plt.title('Residuals vs fitted values plot for homoscedasticity
check')
```



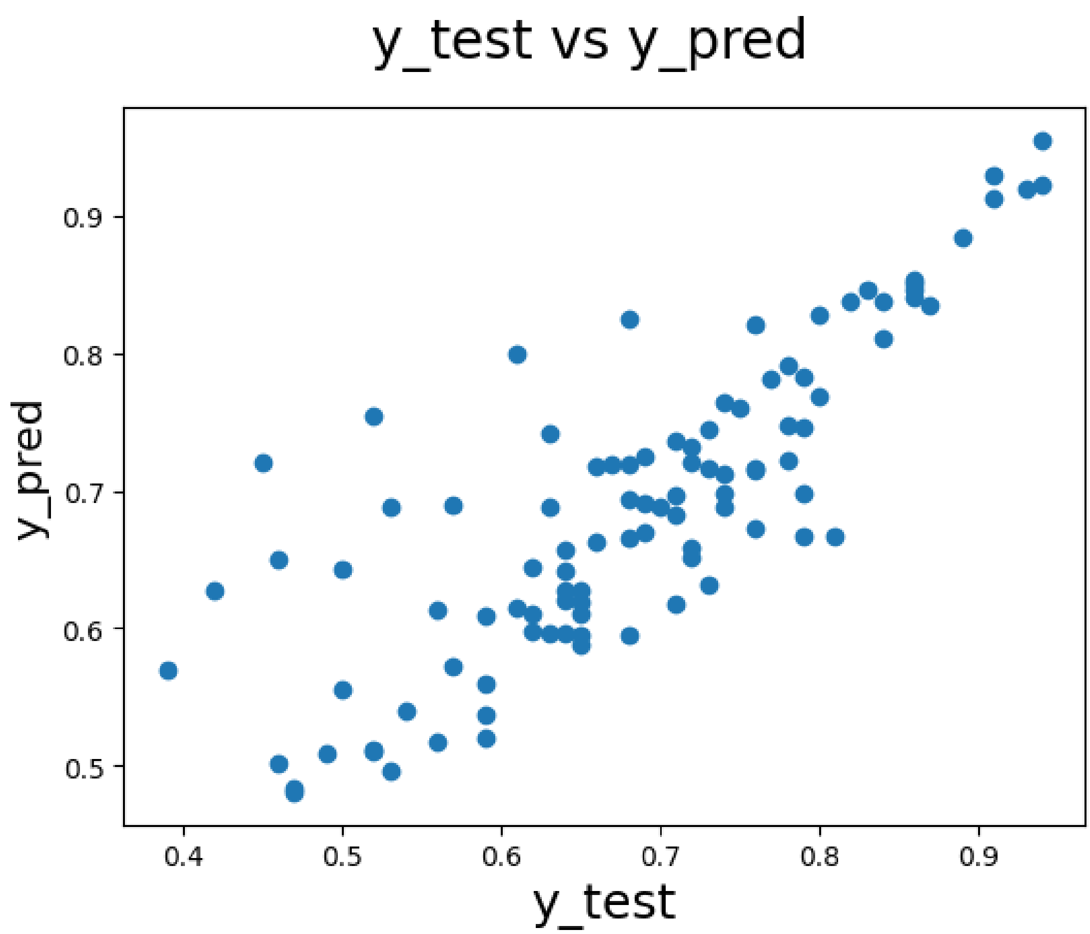

```
import statsmodels.stats.api as sms
from statsmodels.compat import lzip
name = ['F statistic', 'p-value']
test = sms.het_goldfeldquandt(residuals, X_test)
lzip(name, test)

[('F statistic', 0.6722164597747095), ('p-value', 0.8988832419698937)]
```

Here, `null hypothesis` is - error terms are homoscedastic and since p-values >0.05, we fail to reject the null hypothesis

#Model performance evaluation

##On Testing data

```
# Plotting y_test and y_pred to understand the spread.
fig = plt.figure()
plt.scatter(y_test.values, pred)
fig.suptitle('y_test vs y_pred', fontsize=20)              # Plot
heading
plt.xlabel('y_test', fontsize=18)                          # X-label
plt.ylabel('y_pred', fontsize=16)                          # Y-label

Text(0, 0.5, 'y_pred')
```




```
MAE = np.mean(y_test - pred)
MAE

-0.006628840611761088

MSE = np.mean(np.square(y_test - pred))
MSE

0.005318868466346557
```

```
RMSE = np.sqrt(MSE)
RMSE

0.07293057291936322

SS_res = np.mean(np.square(y_test - pred))
SS_total = np.mean(np.square(y_test - y_test.mean()))

r2_score = 1 - (SS_res/SS_total)
r2_score

0.6587167672866537

n, d = X_test_new.shape

adj_r2 = 1 - ((1-r2_score)*(n-1))/(n-d-1)
adj_r2

0.6327495647975948
```

Bias-Variance Tradeoff

Bias is as a result of over simplified model assumptions Variance occurs when the assumptions are too complex The more preferred model is one with low bias and low varinace.

Dimensionality reduction and feature selection can decrease variance by simplifying models.

Similarly, a larger training set tends to decrease variance.

For reducing Bias: Change the model, Ensure the date is truly representative(Ensure that the training data is diverse and represents all possible groups or outcomes.), Parameter tuning.

The bias–variance decomposition forms the conceptual basis for regression regularization methods such as Lasso and ridge regression.

Regularization methods introduce bias into the regression solution that can reduce variance considerably relative to the ordinary least squares (OLS) solution.

Although the OLS solution provides non-biased regression estimates, the lower variance solutions produced by regularization techniques provide superior MSE performance.

Linear and Generalized linear models can be regularized to decrease their variance at the cost of increasing their bias.

#INSIGHTS AND RECOMMENDATIONS

Insights:

1. There are 7 features in the data out of which 3 are continuous and 4 are categorical but with numeric values so no need to encode them.
2. There are no missing values, no duplicated records, no outliers present.
3. Correlation between the features is high so chance of multi-collinearity is high.

4. GRE and TOFEL scores are highly correlated. CGPA is not that correlated with LOR, Research plays huge role for higher LOR ratings. Similarly, CGPA is correlated highly with SOP as higher CGPA means higher academic excellence and that contributes to SOP. Usually students with higher CGPA have higher GRE and TOEFL scores as they are smart and hardworking so they perform better in these tests as well although that's not always true. Also, we see that students having lower CGPA can have higher LOR rating if research papers are published.

5. Chance of admit is highly correlated with GRE, TOEFL and CGPA.

6. We can see that the RMSE is almost same for Linear regression model and Ridge model but is high for Lasso we went ahead with Linear Regression.

7. We used stats models library for building the linear regression model as it gives the p-values for features which helps in deciding if features are statistically significant. SOP and University rating have p-values higher than 0.05 so they are not that useful.

8. R2 and Adj R2 are ~0.85 for model trained on training data.

9. MAE and RMSE are almost 0 for values predicted for testing data which is good.

10. We can derive that Linear Regression can be used as all the assumptions of it are held true such as distribution of data is consistent, mean of residuals is 0, residual distribution is normal, homoskedasticity is present, multi-collinearity is removed as VIF value is lesser than 5 for all features after dropping 'SOP'.

11. We can see that our trained Linear Regression model is able to predict more accurately for the y_test values greater than 0.7

Recommendations:

1. Students are recommended to have higher GRE and TOEFL scores as it increases the chance of admit.

2. Students should keep their CGPA consistently high as it helps in getting the admit. Also, it justifies in getting higher SOP rating as well.

3. Recommended to publish research papers as it enhances the LOR rating.

4. It's fine to not give much attention to university rating and SOP rating as they are not that statistically significant in increasing the chance of admit.

5. It's recommended to use hyperparameter tuning to further enhance the performance of the model.

6. We can try Ridge, Lasso or Elastic Net models to see if there's any improvement in the performance

7. We can check the model performance by dropping the 'University Rating' feature as it's having p-value greater than 0.05. We did not drop that in the solution as it's usually important for predictions.