

**THIRUVALLUVAR UNIVERSITY PERIYAR
ARTS COLLEGE CUDDALORE
– 607001.**



DEPARTMENT OF COMPUTER APPLICATIONS

MACHINE LEARNING WITH PYTHON

Project Title : Intelligent Admissions: The Future of University Decision

Making with Machine Learning

Team Id : NM2023TMID35505

Team Leader : JAYARAJ C(tvu10520CA23)

Team member: NARENDIRAN P(tvu10520CA31)

Team member: YUVARAJ R(tvu10520CA50)

Team member: HARIDOSS P(tvu10520CA19)

Introduction

Overview

University admission is the process by which students are selected to attend a college or university. The process typically involves several steps, including submitting an application, taking entrance exams, and participating in interviews or other evaluations.

Students are often worried about their chances of admission in University. the university admission process for students can be demanding, but by being well-informed, prepared, and organized, students can increase their chances of being admitted to the university of their choice.

The aim of this project is to help students in short listing universities with their profiles. Machine learning algorithms are then used to train a model on this data, which can be used to predict the chances of future applicants being admitted.

With this project, students can make more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

Purpose

This project can help students to predict their chance of admission in universities. The project is employed with a well trained model which can predict accurate results. With proper user(student) input, the flask web app can predict the result with the trained model.

Problem Definition and Design Thinking

Empathy Map

University Admission Prediction

Enter your details and get probability of your admission

Enter GRE score

Enter TOEFL score

Select University No :

☐ 1
☒ 2
☐ 3
☐ 4
☐ 5

Enter SOP

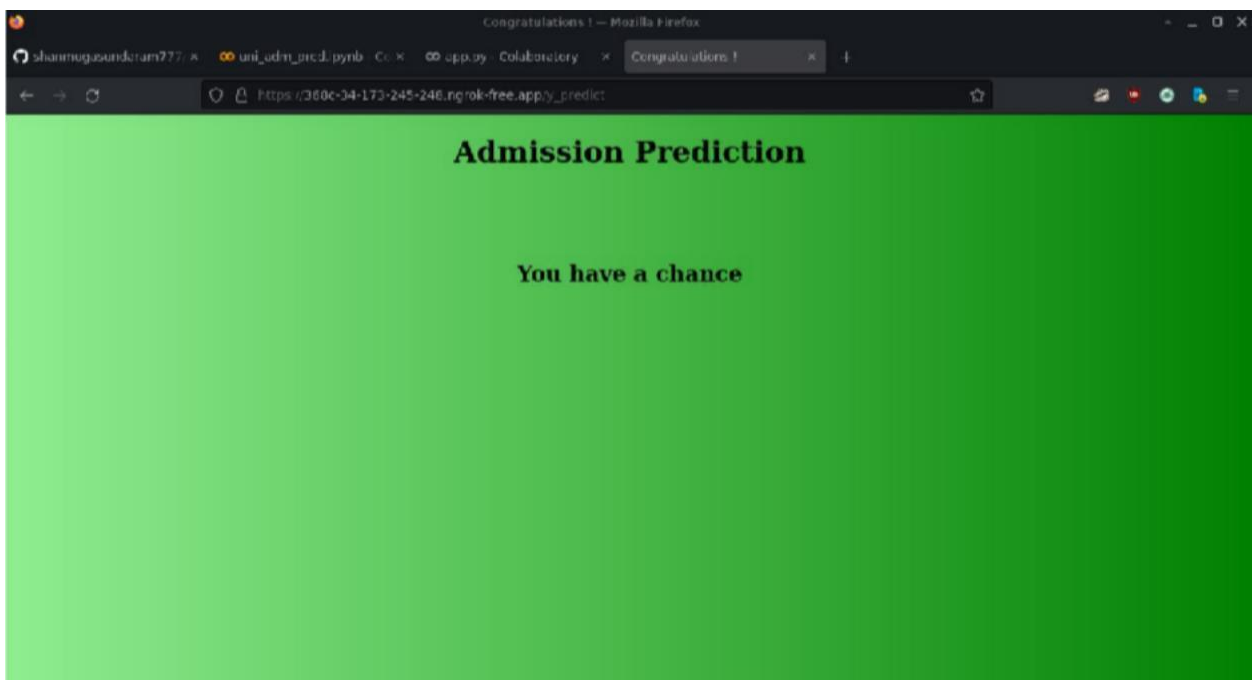
Enter LOR

Enter CGPA

Research

☐ Research
☒ No Research

Prediction



"Chance of Admit" depends on CGPA, GRE, TOEFL. The columns SOP, LOR and Research have less impact on university admission. GRE score, TOEFL score, and CGPA all are linearly related to each other.

Students in research score high in TOEFL and GRE compared to non-research candidates.

Advantages and Disadvantages

Advantages

The project takes into account of all the necessary variables that determine the admission of students in universities.

The variables are absolutely bare minimal and is strictly required to perform accurate predictions.

It helps students to get a preliminary prediction of how their profile/score may perform on the university prediction process.

It enables students to get their overall image on university shortlistings...

Disadvantages

This project only takes into account of minimal variables and special edge cases are not considered.

It omits certain special cases which the students and universities may have come across.

The dataset used is merely adequate not dense enough to train high or world class models that is capable of predicting complex inputs while producing accurate results.

Currently, this project makes use of web technologies such as Google Colab and Ngrok.

Due to poor hardware at our end we have used the web technologies, to deploy the code in other environments and produce a valid webapp, the code must be modified to be run locally on the development server.

Applications

This project unfortunately can only be applied to Education Fields especially, Universities.

And it is only useful for students to assess their profile performance.

The project may be further modified to fit other educational fields such as schools and other educational bodies, that assess student's past performance to allow them in their institutions.

Conclusion

This project is very useful for students to assess their overall profile performance that is necessary to get a overall preliminary of their profile before applying applications to various universities.

It helps them to filter out universities that fit their profile and apply to selected universities that has high rate of being accepted.

Futute Scope

Currently, the project uses a bare minimal dataset that has entries of about 400 rows.

Although this dataset is enough, it is not feature rich.

To further enhance the project, a more dense and feature rich dataset is to be used.

The additional complex features helps in addressing edge or special cases that the students and universities may encounter during their academical span.

Additional algorithms will be used to proof the accuracy of the dataset split. And feature rich data helps in creating and training a model that can generate much precise and accurate results while handling special or edge cases.

The code needs to be further enhanced to get additional variable inputs from user that are helpful in overall prediction with much specialised and accurate results.

Appendix

Source code

University Admission Prediction.ipynb

Machine_Learning_Projects_Vid x Home Page - Select or create a i x University Admission Prediction x +

localhost:8888/notebooks/University%20Admission%20Prediction.ipynb

Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

jupyter University Admission Prediction Last Checkpoint: Last Thursday at 1:39 PM (autosaved)

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel) O

Import Necessary Libraries

Let us import necessary libraries to get started!

```
In [1]: #import necessary libraries
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```

The Data

Let's start by reading in the Admission_Predict.csv file into a pandas dataframe.

```
In [2]: #read_csv is a pandas function to read csv files
data = pd.read_csv("Admission_Predict.csv")

In [3]: #head() method is used to return top n (5 by default) rows of a DataFrame or series.
data.head()
```

```
Out[3]:
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.78
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65
5	6	330	115	5	4.5	3.0	9.34	1	0.90
6	7	321	109	3	3.0	4.0	8.70	1	0.75

Machine_Learning_Projects_Vid x Home Page - Select or create a i x University Admission Prediction x +

localhost:8888/notebooks/University%20Admission%20Prediction.ipynb

Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

jupyter University Admission Prediction Last Checkpoint: Last Thursday at 1:39 PM (autosaved)

File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel) O

```
In [4]: #let us drop Serial No. Column as it is not required for prediction
data.drop(["Serial No."],axis=1,inplace=True)
data.head()
```

```
Out[4]:
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	337	118	4	4.5	4.5	9.65	1	0.92
1	324	107	4	4.0	4.5	8.87	1	0.78
2	316	104	3	3.0	3.5	8.00	1	0.72
3	322	110	3	3.5	2.5	8.67	1	0.80
4	314	103	2	2.0	3.0	8.21	0	0.65

#describe() method computes a summary of statistics like count, mean, standard deviation, min, max and quartile values.

```
In [5]: data.describe()
```

```
Out[5]:
```

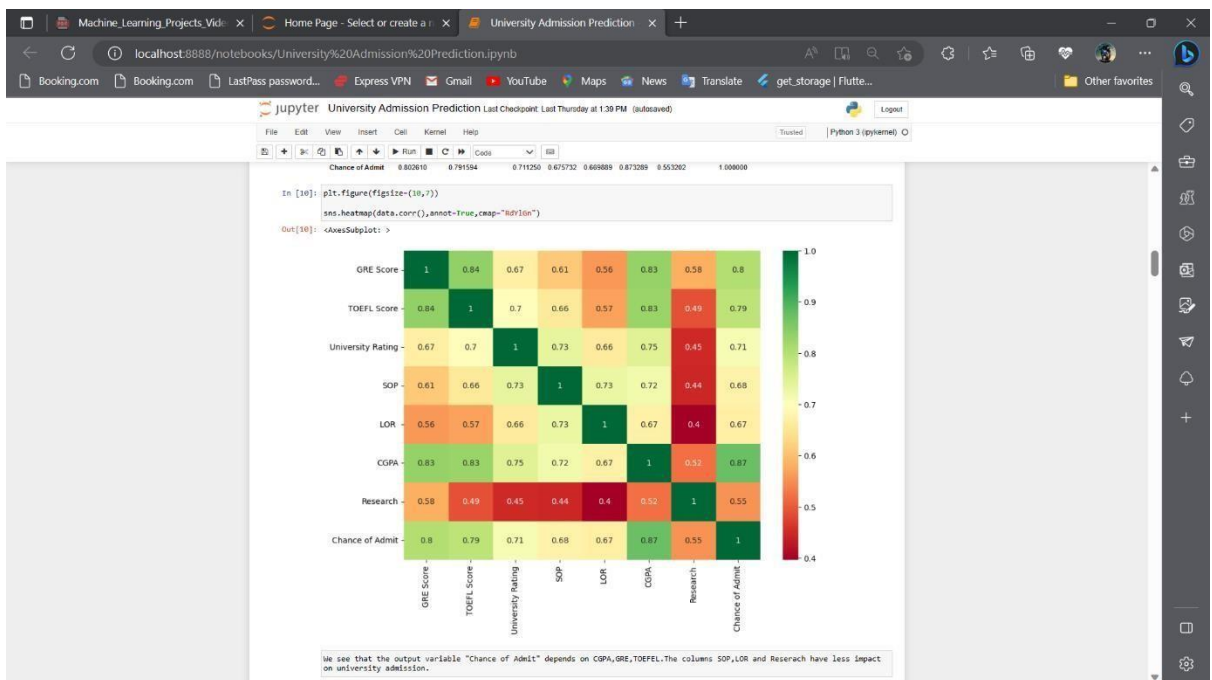
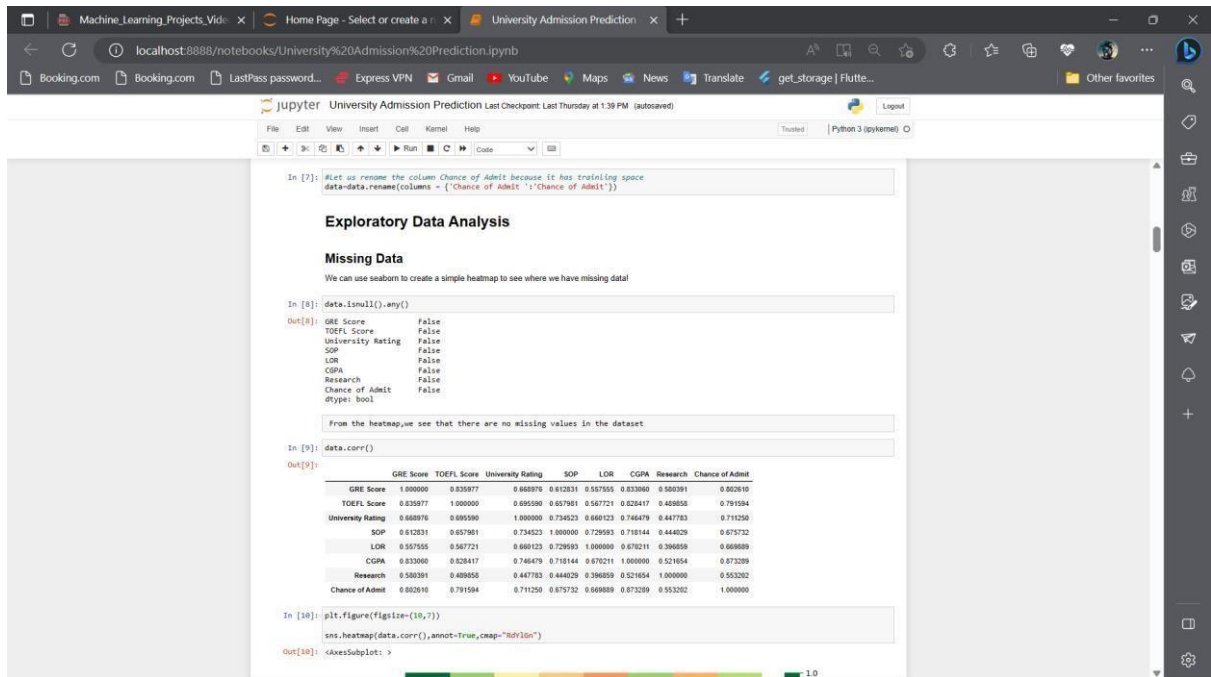
	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.307500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.088514	1.143728	1.006689	0.886478	0.596317	0.498362	0.142609
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	306.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.002000	1.000000	0.820000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

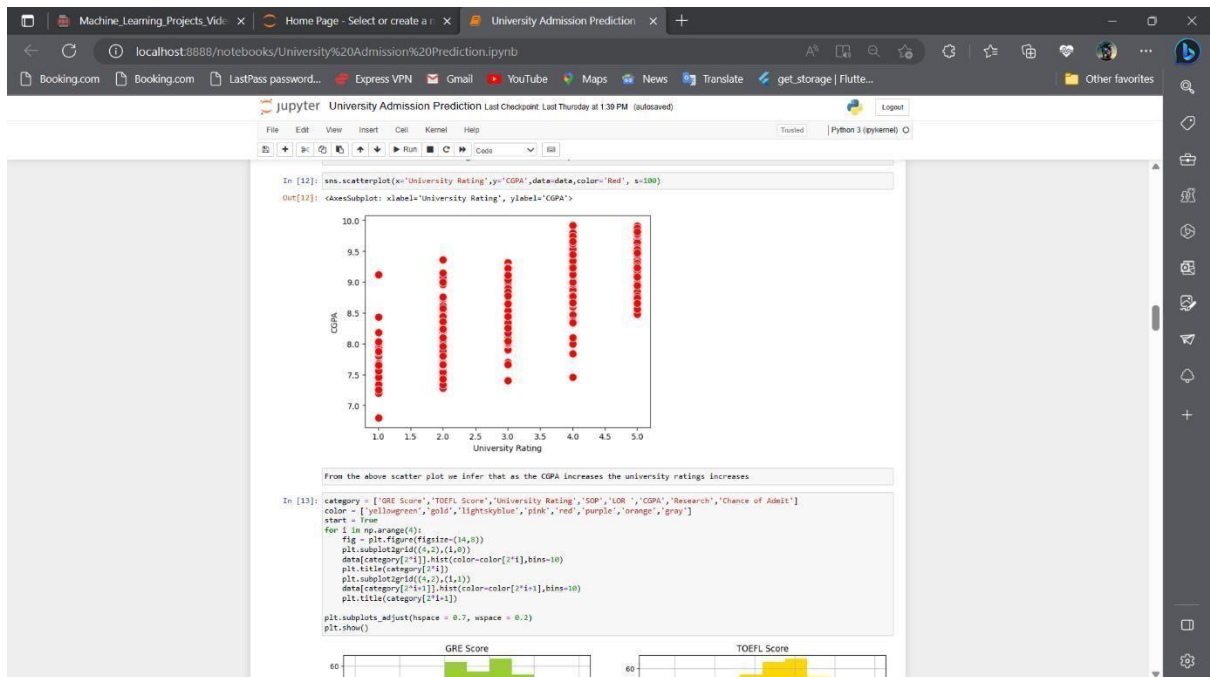
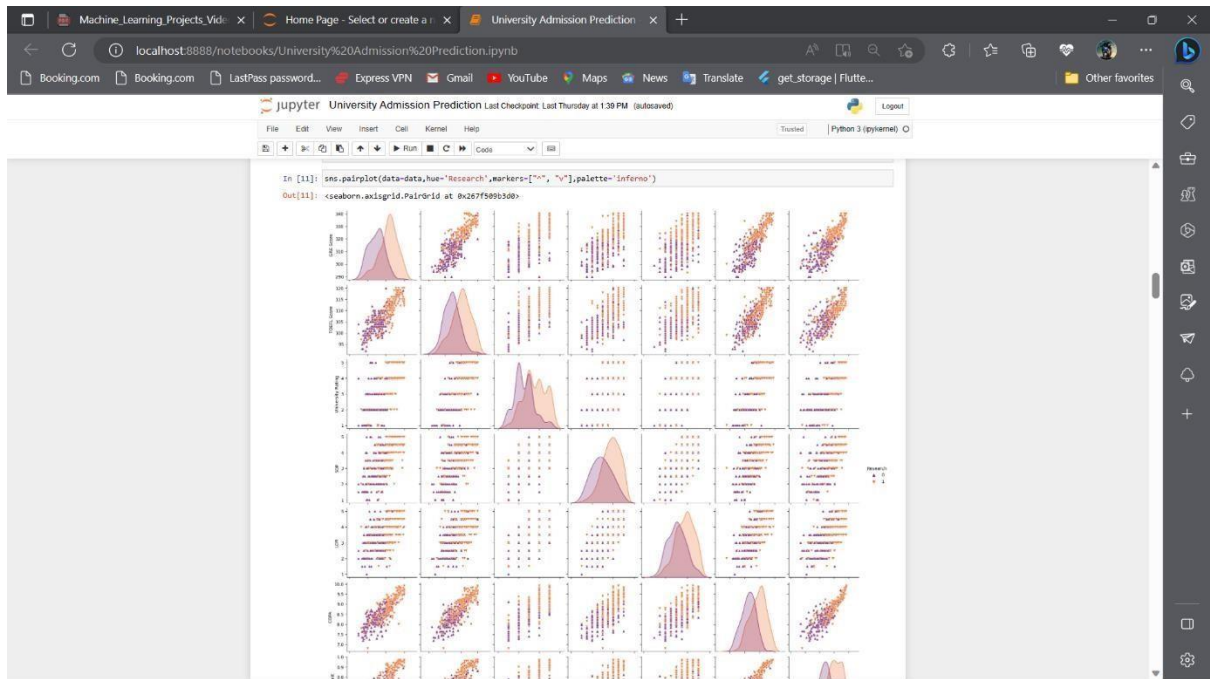
From the data we infer that there are only decimal values and no categorical values

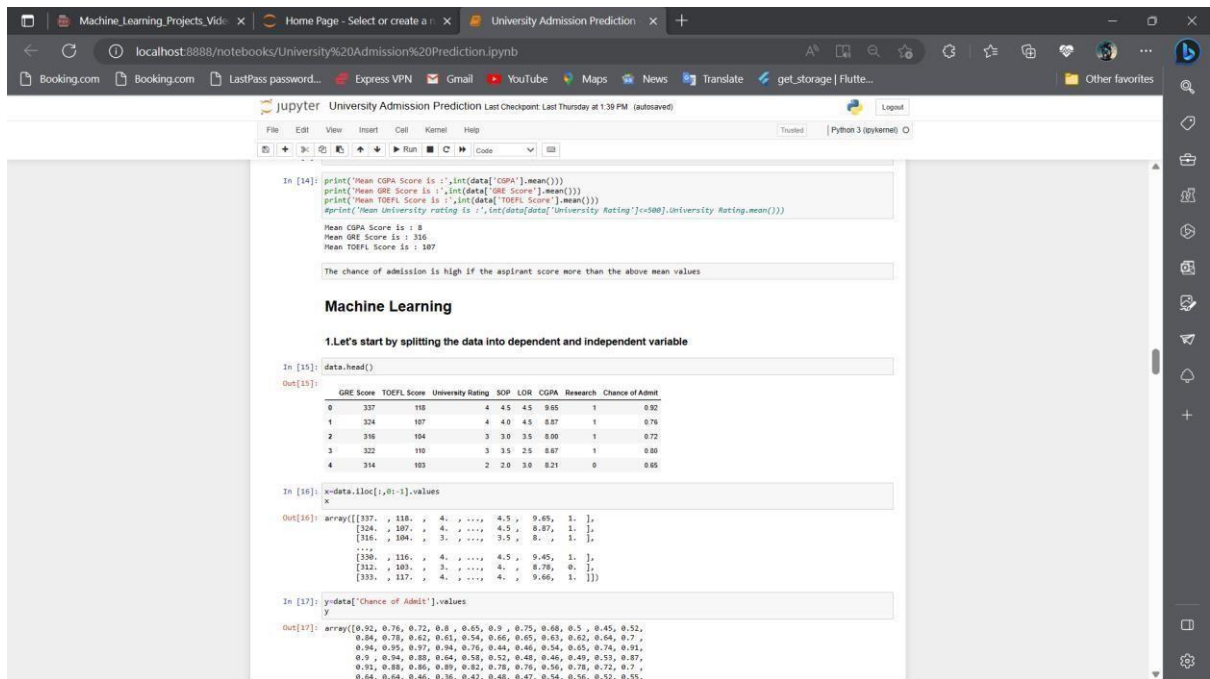
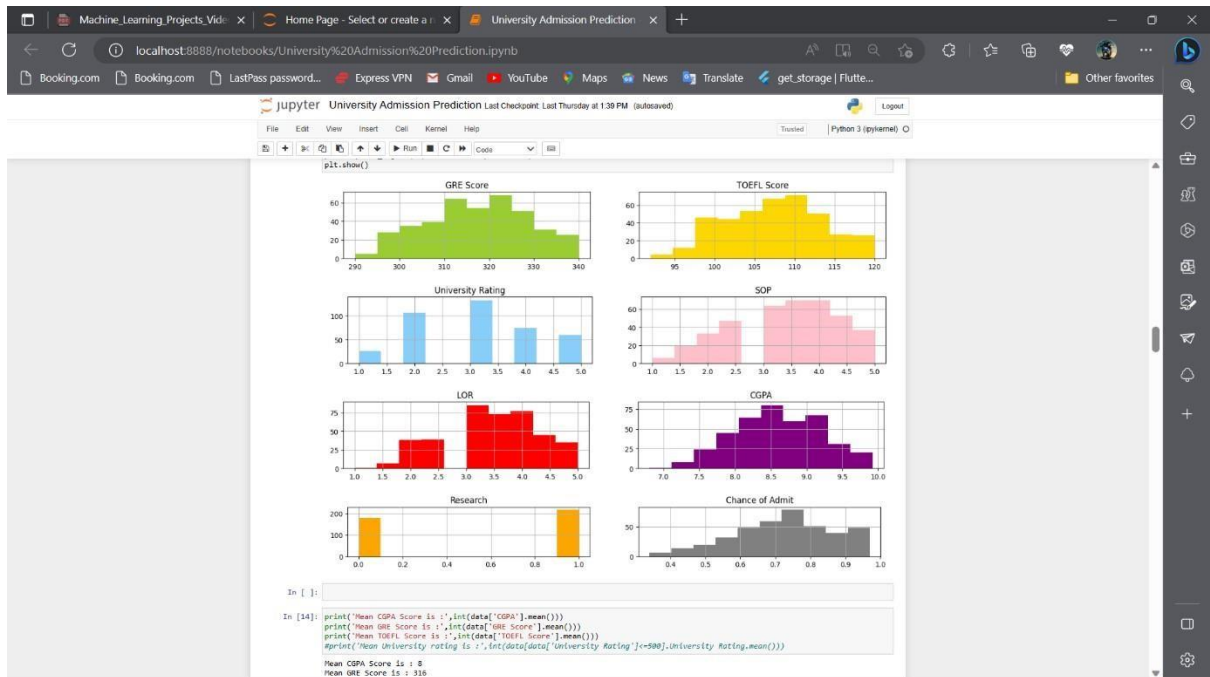
```
In [6]: data.info()
```

```
Out[6]:
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   GRE Score           400 non-null    int64
 1   TOEFL Score         400 non-null    int64
 2   University Rating    400 non-null    int64
 3   SOP                 400 non-null    float64
 4   LOR                 400 non-null    float64
 5   CGPA                400 non-null    float64
 6   Research             400 non-null    int64
 7   Chance of Admit     400 non-null    float64
```








```
Machine_Learning_Projects_Vid... x Home Page - Select or create a... x University Admission Prediction x +
localhost:8888/notebooks/University%20Admission%20Prediction.ipynb
Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

Jupyter University Admission Prediction Last checkpoint: Last Thursday at 1:30 PM (autosaved)
Python 3 (ipykernel)

In [41]:
ann_pred = classifier.predict(x_test)
ann_pred = (ann_pred>0.5)
print(accuracy_score(ann_pred,y_test))
print("****ADM Model****")
print("Confusion Matrix")
print(confusion_matrix(y_test,ann_pred))
print("Classification Report")
print(classification_report(y_test,ann_pred))

3/3 [=====] - 0s 0s/step
0.125
****ADM Model****
Confusion Matrix
[[18  0]
 [70  0]]
Classification Report
              precision    recall  f1-score   support

 False      0.12      1.00      0.22      10
  True      0.00      0.00      0.00      70

 accuracy      0.12      0.12      0.12      80
 macro avg      0.06      0.50      0.11      80
 weighted avg      0.02      0.12      0.03      80

C:\Users\VP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\VP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\VP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

In [42]:
#Testing on test & random input values
print("Predicting on test input")
ann_pred = classifier.predict(x_test)
ann_pred = (ann_pred>0.5)
print("output is: ",ann_pred)
print("Predicting on random input")
ann_pred_om = classifier.predict(sc.transform([[137,118,4,4.5,4.5,9.05,1]]))
ann_pred_om = (ann_pred_om>0.5)
print("output is: ",ann_pred_om)

Predicting on test input
3/3 [=====] - 0s 0s/step
output is: [[False]]
```

```
Machine_Learning_Projects_Vid... x Home Page - Select or create a... x University Admission Prediction x +
localhost:8888/notebooks/University%20Admission%20Prediction.ipynb
Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

Jupyter University Admission Prediction Last checkpoint: Last Thursday at 1:30 PM (autosaved)
Python 3 (ipykernel)

In [43]:
ann_pred_train = classifier.predict(x_train)
ann_pred_train = (ann_pred_train>0.5)
print(accuracy_score(ann_pred_train,y_train))
print("****ADM Model****")
print("Confusion Matrix")
print(confusion_matrix(ann_pred_train,y_train))
print("Classification Report")
print(classification_report(ann_pred_train,y_train))

18/18 [=====] - 0s 889us/step
0.078125
****ADM Model****
Confusion Matrix
[[ 25 205]
 [  0  0]]
Classification Report
              precision    recall  f1-score   support

 False      1.00      0.00      0.14     320
  True      0.00      0.00      0.00      20

 accuracy      0.50      0.04      0.07     320
 macro avg      0.50      0.04      0.07     320
 weighted avg      1.00      0.00      0.14     320

C:\Users\VP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\VP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\VP\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

In [44]: pickle.dump(lr,open('university.pkl','wb'))

In [ ]:
```



```
File Edit Selection View Go Run Terminal Help • app.py - Visual Studio Code
C:\Users\HP\Desktop\COMPLETED\Project 2\Flask> app.py y_predict
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4 app = Flask(__name__)
5 model = pickle.load(open('university.pkl', 'rb'))
6 @app.route('/')
7 def home():
8     return render_template('Demo2.html')
9 @app.route('/y_predict', methods=['POST'])
10 def y_predict():
11     """
12     For rendering results on HTML GUI
13     """
14     # min max scaling
15     min1 = [290.0, 92.0, 1.0, 1.0, 1.0, 6.8, 0.0]
16     max1 = [340.0, 120.0, 5.0, 5.0, 5.0, 9.92, 1.0]
17     k = [float(x) for x in request.form.values()]
18     p = []
19     for i in range(7):
20         l = (k[i]-min1[i])/(max1[i]-min1[i])
21         p.append(l)
22     prediction = model.predict([p])
23     print(prediction)
24     output = prediction[0]
25     if (output == False):
26         return render_template('noChance.html', prediction_text='You Dont have a chance of getting admission')
27     else:
28         return render_template('chance.html', prediction_text='You have a chance of getting admission')
29 if __name__ == "__main__":
30     app.run(debug=False)
31
```