

PROJECT REPORT

Project name: University Admit Eligibility Predictor Using IBM Watson

Table Of Contents:

1. Introduction
 - 1.1 Overview
 - 1.2 Purpose
2. Literature Survey
 - 2.1 Existing problem
 - 2.2 Proposed solution
3. Theoretical Analysis
 - 3.1 Block diagram
 - 3.2 Hardware / Software designing
4. Experimental Investigations
5. Flowchart
6. Result
7. Advantages & Disadvantages
8. Applications
9. Conclusion
10. Future scope
11. Bibliography
- Appendix
 - a. Source code
 - b. UI output screenshot

Submitted by,

Merlin Johnson

1. INTRODUCTION

1.1 Overview:

The project is implemented using a Machine-Learning model that predicts whether the user is eligible for an admission in the selected rated universities with provided details such as marks and others. The algorithm works in such a way that when the user provides the details such as (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research) the percentage of chance of admit is displayed.

The user is provided with a UI (Web based application) in which the user can enter the details mentioned above for prediction. The main advantage of this is that the user can avoid long process of having to check the eligibility of a university admission by himself and make use of this application to predict the eligibility / chance of admit.

1.2 Purpose:

The purpose of this project is to make the prediction of eligibility of an admission to a rated university with ease using a UI with the provided user details (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research). This also eliminates the possibility of human errors.

2. LITERATURE SURVEY

2.1 Existing problem:

Previous research done in this area used Naive Bayes algorithm which will evaluate the success probability of student application into a respective university but the main drawback is they didn't consider all the factors which will contribute in the student admission process like TOEFL/IELTS, SOP, LOR and under graduate score.

Bayesian Networks Algorithm have been used to create a decision support network for evaluating the application submitted by foreign students of the university. This model was developed to forecast the progress of prospective students by comparing the score of students currently studying at university. The model thus predicted whether the aspiring student should be admitted to university on the basis of various scores of students. Since the comparisons are made only with students who got admission into the universities but not with students who got their admission rejected so this method will not be that much accurate.

2.2 Proposed solution:

These problems can be resolved by using regression algorithms / classification algorithms as they can consider most of the features for prediction. Linear regression / KNN classification / Random forest regression can be used as the machine learning model for the model. XG boost model can also be used which performs better on small to medium scale datasets but the model giving accurate and desired results only will be selected.

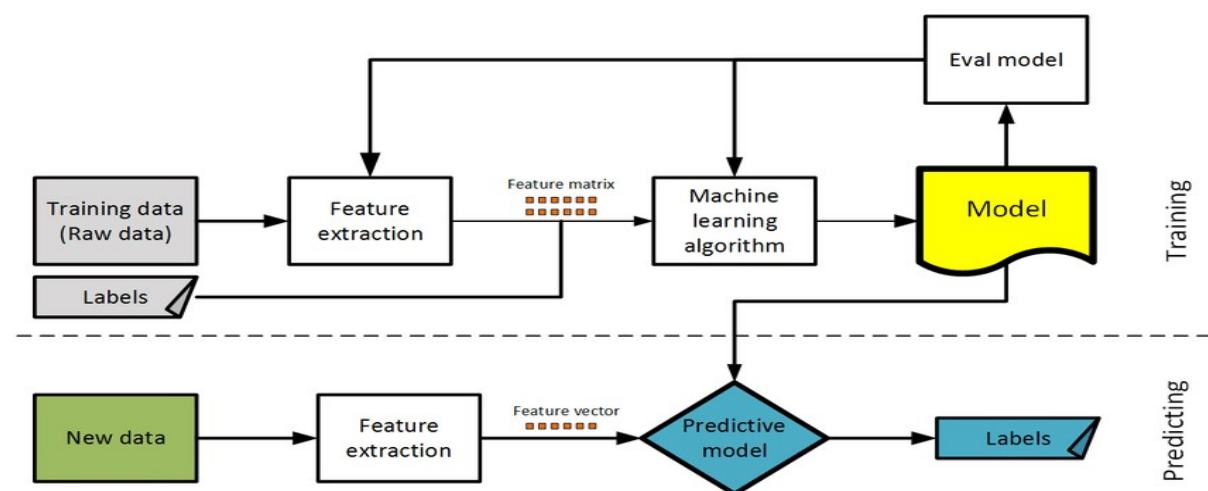
The aim of the proposed system is to address the limitations of the current system.

The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system:

- Reach to geographically scattered student.
- Reducing time in activities
- Paperless admission with reduced man power
- Operational efficiency

3.THEORETICAL ANALYSIS

3.1 Block diagram:



3.2 Hardware/Software designing:

ITEM	CONTENT
CPU	Intel i3 7th gen or above/ AMD Ryzen 3500 or above
GPU	Integrated / Dedicated (atleast 2GB)
RAM	8 GB
Operating system	Windows 10/ Linux / MacOS
Programming language	Python 3.9
ML library	Scikit learn

Other libraries	Matplot,seaborn,numpy,pandas,pickle
-----------------	-------------------------------------

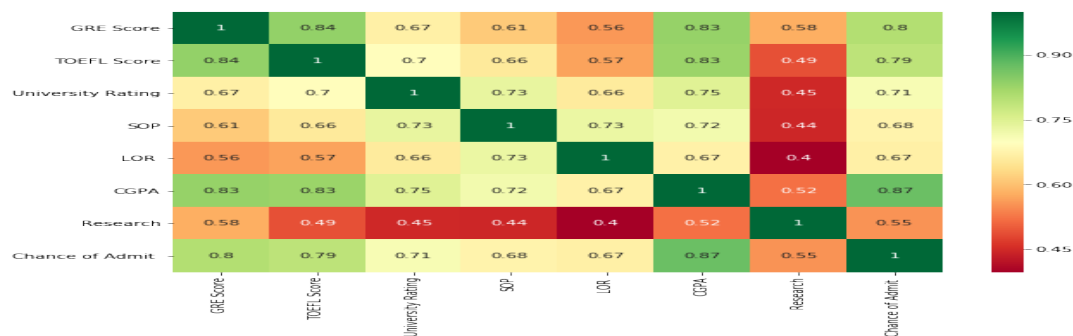
SOFTWARE DESIGNING:

1. Jupyter Notebook Environment
2. Spyder
3. Machine Learning Algorithms
4. Python (pandas, numpy, matplotlib, seaborn, sklearn)
5. HTML
6. Flask

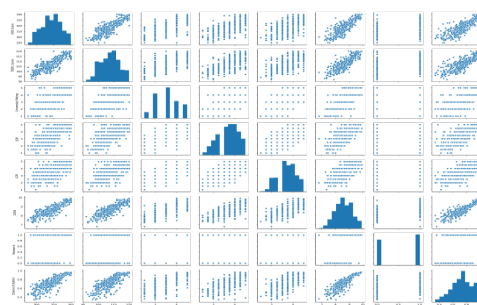
We developed this Visa Approval status prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a micro frame work because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

4. EXPERIMENTAL ANALYSIS

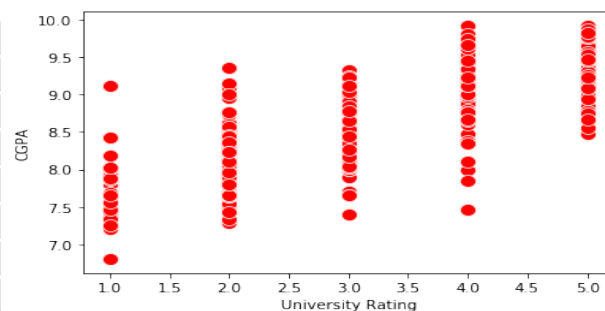
Heat map of the data



Pair plot



Scatter plot

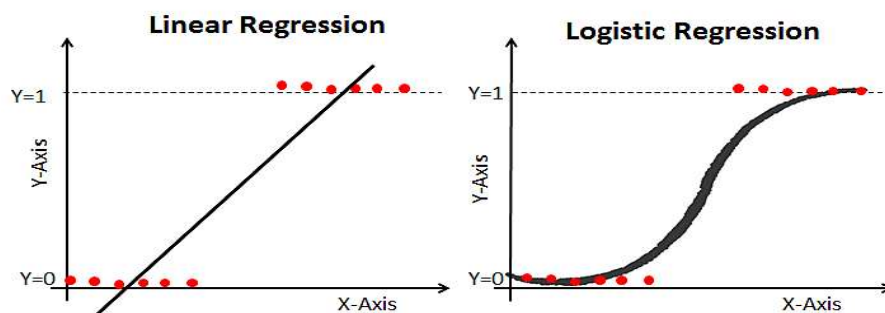


Logistic Regression:

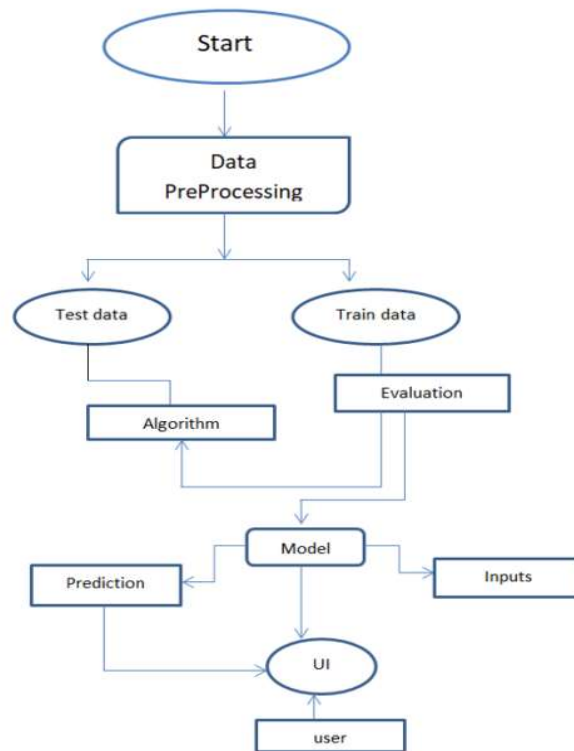
This type of statistical model(also known as *logit model*) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.

Linear Regression:

Linear regression models are used to identify the relationship between a continuous dependent variable and one or more independent variables. When there is only one independent variable and one dependent variable, it is known as simple linear regression, but as the number of independent variables increases, it is referred to as multiple linear regression. For each type of linear regression, it seeks to plot a line of best fit through a set of data points, which is typically calculated using the least squares method.

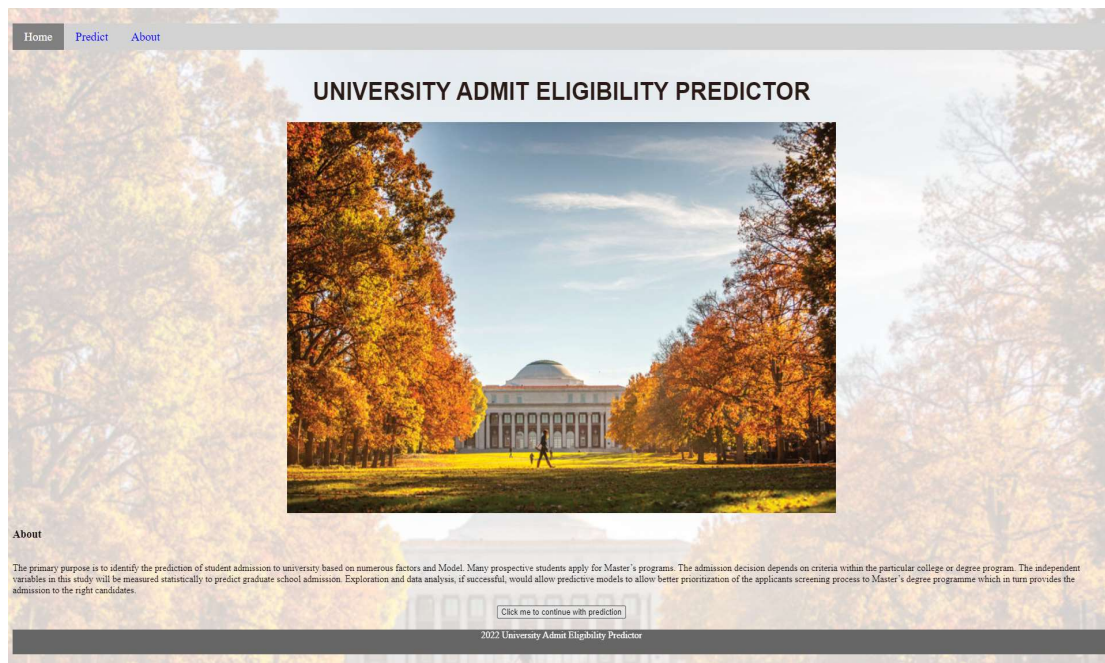


5. FLOWCHART



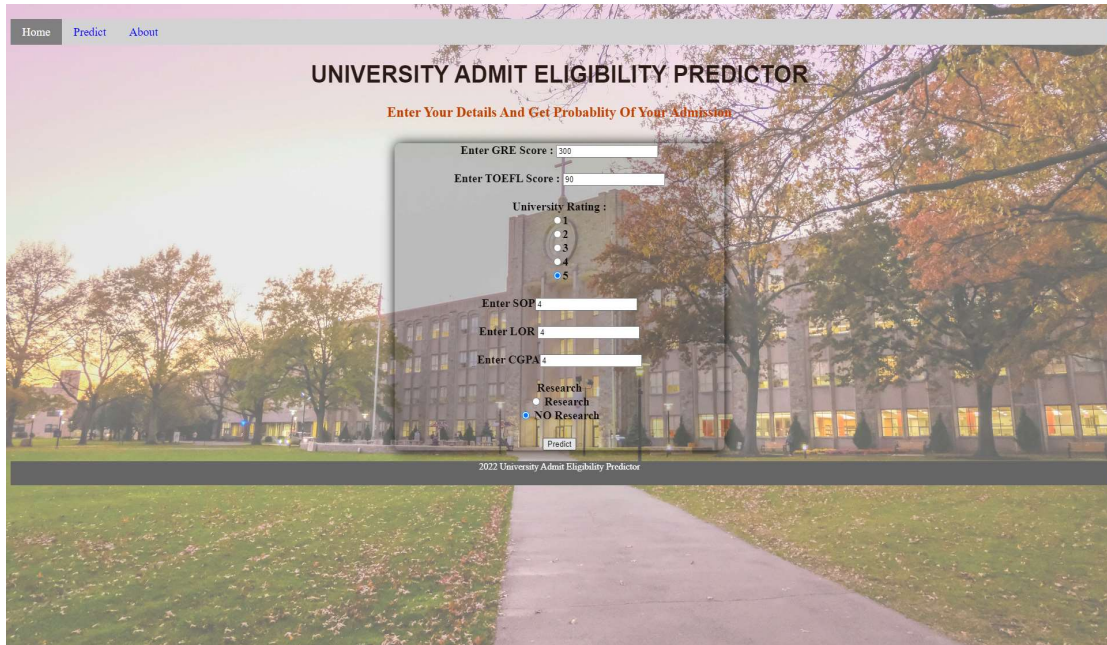
6. RESULT

Home page



Demo2.html

For getting admission in reputed university for master's degree. We will enter our results of all the fields enter here and check the probability here:



The screenshot shows a web application titled "UNIVERSITY ADMIT ELIGIBILITY PREDICTOR" with a navigation bar containing "Home", "Predict", and "About". Below the title is a subtitle: "Enter Your Details And Get Probability Of Your Admission". The form includes the following fields and options:

- Enter GRE Score :
- Enter TOEFL Score :
- University Rating :
 - ☐ 1
 - ☐ 2
 - ☐ 3
 - ☐ 4
 - ☒ 5
- Enter SOP :
- Enter LOR :
- Enter CGPA :
- Research :
 - ☐ Research
 - ☒ NO Research
-

At the bottom of the form, it says "© 2022 University Admit Eligibility Predictor". The background of the page is a photograph of a university campus with trees and a building.

Chance.html

→If student getting good results prediction shows your eligible.

Predicting Chance of Admission

A machine learning Web App, Built with Flask

Prediction: You have a chance

CONGRATULATIONS



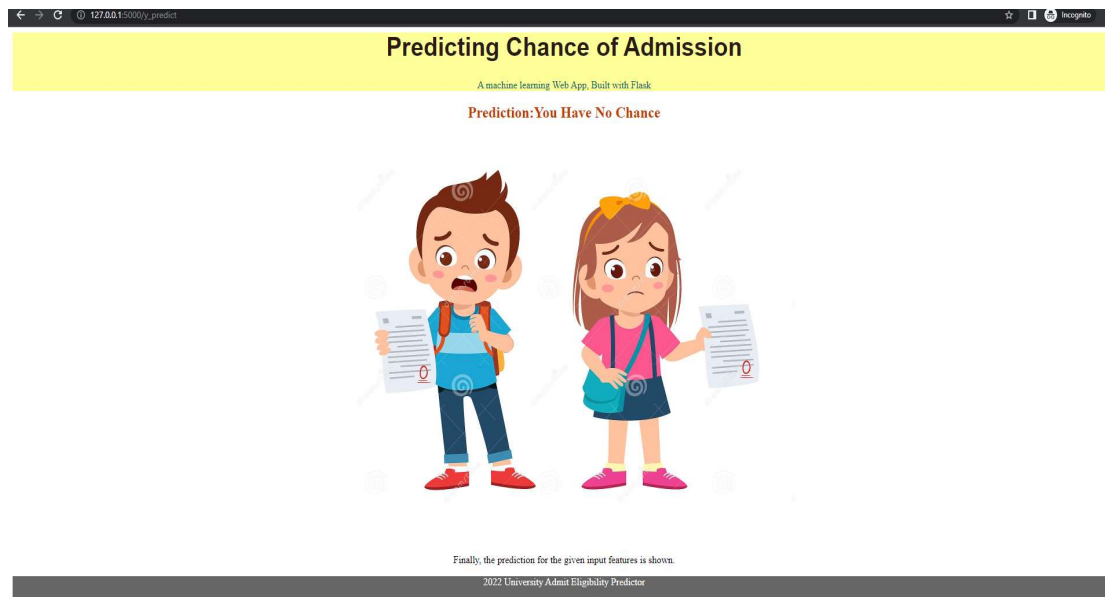
Finally, the prediction for the given input features is shown.

2022 University Admit Eligibility Predictor

A screenshot of a web browser displaying the 'University Admit Eligibility Predictor' application. The browser's address bar shows '127.0.0.1:5000/assessment/'. The page has a navigation bar with 'Home', 'Predict', and 'About' links. The main heading is 'UNIVERSITY ADMIT ELIGIBILITY PREDICTOR' with a sub-heading 'Enter Your Details And Get Probability Of Your Admission'. The form contains several input fields: 'Enter GRE Score' (110), 'Enter TOEFL Score' (45), 'University Rating' (radio buttons 1-5, with 5 selected), 'Enter SOP' (2), 'Enter LOR' (3), and 'Enter CGPA' (4). There are also radio buttons for 'Research' and 'NO Research' (selected), and a 'Predict' button. The footer of the form area says '2022 University Admit Eligibility Predictor'. The browser's taskbar at the bottom shows the date and time as 10:08 AM on 15-10-2022.

[noChance.html](#)

→ If student is not eligible it shows you are not eligible based on the results only.



7. ADVANTAGES & DISADVANTAGES

ADVANTAGES

It helps student for making decision for choosing a right college.

- Here the chance of occurrence of error is less when compared with the existing system.
- Avoids data redundancy and inconsistency.
- It is fast, efficient and reliable.

DISADVANTAGES

- Machine errors are unavoidable when occurred. (Hardware failure, network failure, others).
- The predictions made are not 100% accurate but accurate to an acceptable value.
- Data Acquisition, Time and Space, Internet Issues

8. APPLICATIONS

- Reach to geographically scattered student.
- Reducing time in activities
- Paperless admission with reduced man power
- Operational efficiency

9. CONCLUSION

The project uses a Logistic regression to predict the output and a web application is built to make the UI more accessible and easy using various technologies such as python, HTML5, CSS, Flask, Scikit, Matplot, Numpy, Pandas, Seaborn and other libraries. After the deployment of the web application, it can be accessed from anywhere with internet connection. This project reduces the long hours of analysis to predict the eligibility of the admission to a rated university. The subject of this

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

GRE Score
TOEFL Score
University Rating
SOP
LOR
CGPA

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

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

10. FUTURE SCOPE

University Admission depends on many factors, among them GPA, GRE and TOEFL are most important. We have used these three features in this project but more features can be implemented to get more accurate result. Other features could be Statement of Purpose (SOP), Letter of Recommendation (LOR), industry experience, internship experience, papers published, journals published etc. Also, as an extension to this work, recommendation of university with respect to research interest can be made with further study.

As mentioned before the data we actually fed to model is far less than the total data set as we first filter it based on user's intended major . So, we are working on insufficient dataset. Even though the accuracy is ranged between 70 % - 85%. the result will never be satisfied to the user. So, we can work on huge data set. We can also add more features to our system like University ranking, University News feed. This project is focused for the students intending to pursue master and PhD degrees, we can use dataset of undergraduate college admission and use it for the students intending to pursue bachelor degree.

The future scope of this project is very broad. Few of them are:

- This can be implemented in less time for proper admission process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.

11. BIBLIOGRAPHY

<https://towardsdatascience.com/predicting-h-1b-status-using-random-forest-dc199a6d254c>

<https://smartinternz.com/guided-project/visa-approval-prediction>

APPENDIX

a. Source code

Python(source Code)

```
University Admit Eligibility Predictor
University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)

In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('whitegrid')

In [2]: data=pd.read_csv("C:\Project\Dataset\Admission_Predict.csv")

In [3]: data.head()
Out[3]:
   Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR  CGPA  Research  Chance of Admit
0           1         337          110                4.0    4.5  4.5  0.05         1         0.92
1           2         324          107                4.0    4.0  4.5  0.07         1         0.76
2           3         316          104                3.0    3.0  3.5  0.00         1         0.72
3           4         322          110                3.5    3.5  2.5  0.07         1         0.80
4           5         314          103                2.0    2.0  3.0  0.21         0         0.65

In [4]: 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          110                4.0    4.5  4.5  0.05         1         0.92
1         324          107                4.0    4.0  4.5  0.07         1         0.76
2         316          104                3.0    3.0  3.5  0.00         1         0.72
3         322          110                3.5    3.5  2.5  0.07         1         0.80
4         314          103                2.0    2.0  3.0  0.21         0         0.65

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   316.507500      107.410000      3.087500      3.400000      3.452500      0.598925      0.547500      0.724350
std    11.473040       6.069514       1.143720      1.000000      0.950476      0.595317      0.498362      0.142009
min    290.000000      92.000000      1.000000      1.000000      1.000000      0.000000      0.000000      0.340000
25%    296.000000      103.000000      2.000000      2.500000      3.000000      0.170000      0.000000      0.640000
50%    317.000000      107.000000      3.000000      3.000000      3.500000      0.510000      0.000000      0.730000
75%    326.000000      112.000000      4.000000      4.000000      4.000000      0.825000      1.000000      0.830000
max    340.000000      120.000000      5.000000      5.000000      5.000000      1.000000      1.000000      0.970000
```

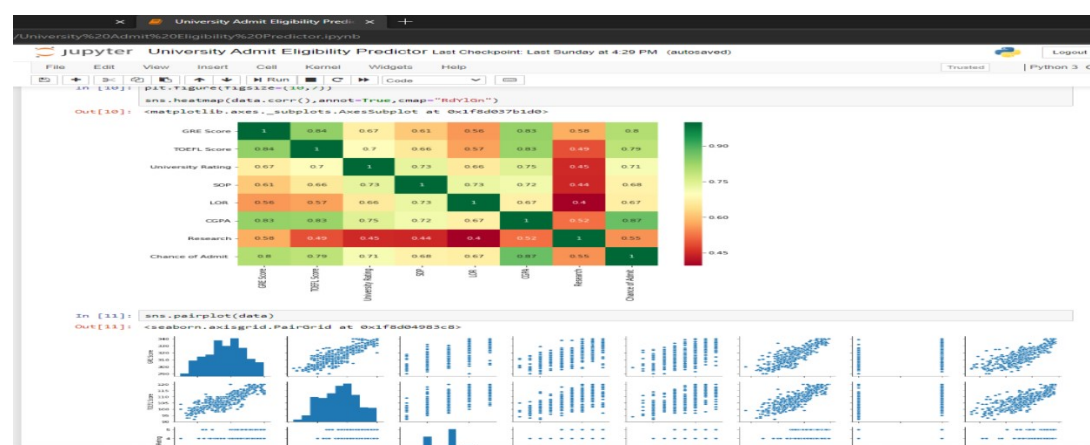
```
University Admit Eligibility Predictor
University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)

In [6]: data.info()
Out[6]:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
GRE Score      400 non-null int64
TOEFL Score    400 non-null int64
University Rating  400 non-null float64
SOP            400 non-null float64
LOR            400 non-null float64
CGPA           400 non-null float64
Research       400 non-null int64
Chance of Admit  400 non-null float64
dtypes: float64(4), int64(4)
memory usage: 25.1 KB

In [7]: data.isnull().any()
Out[7]:
GRE Score      False
TOEFL Score     False
University Rating  False
SOP             False
LOR             False
CGPA            False
Research        False
Chance of Admit  False
dtype: bool

In [8]: data.isnull().sum()
Out[8]:
GRE Score      0
TOEFL Score     0
University Rating  0
SOP             0
LOR             0
CGPA            0
Research        0
Chance of Admit  0
dtype: int64

In [9]: data.corr()
Out[9]:
           GRE Score  TOEFL Score  University Rating  SOP  LOR  CGPA  Research  Chance of Admit
GRE Score    1.000000    0.689976    0.689976    0.689976  0.612831  0.571555  0.833060  0.580391
TOEFL Score  0.689976    1.000000    0.689976    0.689976  0.577221  0.522417  0.489558  0.791594
University Rating  0.689976  0.689976    1.000000    0.734523  0.660123  0.748479  0.447763  0.711250
SOP          0.612831  0.689976    0.689976    1.000000    0.729593  0.718144  0.444029  0.675732
LOR          0.571555  0.577221    0.660123  0.729593    1.000000    0.670211  0.396859  0.669889
CGPA         0.833060  0.522417    0.748479  0.718144  0.670211    1.000000    0.521654  0.873289
Research      0.580391  0.489558    0.447763  0.444029  0.396859    0.521654    1.000000  0.553202
Chance of Admit 0.791594  0.669889    0.711250  0.675732  0.669889  0.873289    0.553202    1.000000
```




```

for i in range(7):
    l = (k[i]-min1[i])/(max1[i]-min1[i])
    p.append(l)
prediction = model.predict([p])
print(prediction)
output = prediction[0]
if output== False:
    return render_template("noChance.html", prediction_text="you dont have a chance")
else:
    return render_template("chance.html", prediction_text="you have a chance")
if __name__ == "__main__":
    app.run(debug=False)

```

Index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>university admission predictor</title>
<link rel="stylesheet" href="../static/css/background.css" />
</head>
<body>
<div class="opacity3"></div>
<div class="big">
<nav>
<ul>
<li><a class="active" href="/home1">Home</a></li>
<li><a href="/assesment">Predict</a></li>
<li><a href="/about">About</a></li>
</ul>
</nav>
<br><h1><center>UNIVERSITY ADMIT ELIGIBILITY PREDICTOR</center></h1>
<div class="p-t-30"><center>

</center> </div>
<h3>About</h3>
<div class="row row-space">
<p>The primary purpose is to identify the prediction of student admission to university based on numerous factors and Model. Many prospective students apply for Master's programs. The admission decision depends on criteria within the particular college or degree program. The independent variables in this study will be measured statistically to predict graduate school admission. Exploration and data analysis, if successful, would allow predictive models to allow better prioritization of the applicants screening process to Master's degree programme which in turn provides the admission to the right candidates.</p>
</div>
<center>
<form action="/assesment">
<button type="submit">Click me to continue with prediction</button></center>
</form>
<br>
<center>
<div class="footer">2022 University Admit Eligibility Predictor</div></center>
</div>
</body>
</html>

```


Demo2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>university admit eligibility predictor</title>
<link rel="stylesheet" href="../static/css/background.css" />
</head>
<body>
<center>
<div class="opacity"></div>
<div class="big">
<nav>
<ul>
<li><a class="active" href="/home1">Home</a></li>
<li><a href="/assesment">Predict</a></li>
<li><a href="/about">About</a></li>
</ul>
</nav>
<h1><B>UNIVERSITY ADMIT ELIGIBILITY PREDICTOR</B></h1>
<h2><B>Enter Your Details And Get Probablity Of Your Admission</B></h2>
<br>
<div class="form1">
<form method="POST" action="{ {url_for('y_predict')}}">
<B> Enter GRE Score : <input type="text" name="t1" placeholder="GRE Score (out of 340)"
required="required"/><br><br>
Enter TOEFL Score : <input type="text" name="t2" placeholder="TOEFL Score (out of 120)"
required="required"/><br><br>
<table for="University Rating"> University Rating : </table><br>
<input type="radio" name="University Rating" value=1>1</input><br>
<input type="radio" name="University Rating" value=2>2</input><br>
<input type="radio" name="University Rating" value=3>3</input><br>
<input type="radio" name="University Rating" value=4>4</input><br>
<input type="radio" name="University Rating" value=5>5</input><br>
<br>
Enter SOP<input type="text" name="sop" placeholder="SOP (out of 5)"
required="required"/><br><br>
Enter LOR <input type="text" name="lor" placeholder="LOR (out of 5)"
required="required"/><br><br>
Enter CGPA<input type="text" name="cgpa" placeholder="CGPA (out of 10)"
required="required"/><br><br>
Research<br> <input type="radio" name="research" value="1" > Research <br>
<input type="radio" name="research" value="0"> NO Research <br><br>
<button type="submit">Predict</button></B>
</form>
</div>
<br>
<div class="footer">2022 University Admit Eligibility Predictor</div></center>
</center>
</div>
</body>
</html>
```

Chance.html

```
<html>
```

```

<head>
<title>university admission predictor</title>
<link rel="stylesheet" href="../static/css/background.css" />
</head>
<body >
<center>
<div>
<div class="h-container">
<h1 class="heading">Predicting Chance of Admission</h1>
<p class="para">A machine learning Web App, Built with Flask</p>
</div>
<div class="container2">
<h2 class="heading2">Prediction:<span class="s-heading">You have a chance</span></h2>


</div><br><br><br><br>
<p class="end-para">Finally, the prediction for the given input features is shown.</p>
</div><br><br><br>
<div class="footer">2022 University Admit Eligibility Predictor</div>
</center>
</body>
</html>

```

noChance.html

```

<html>
<head>
<link rel="stylesheet" href="../static/css/background.css" />
</head>
<body>
<center>
<div>
<div class="h-container">
<h1 class="heading">Predicting Chance of Admission</h1>
<p class="para">A machine learning Web App, Built with Flask</p>
</div>
<div class="container2">
<h2 class="heading2">Prediction:<span class="s-heading">You Have No Chance</span></h2>

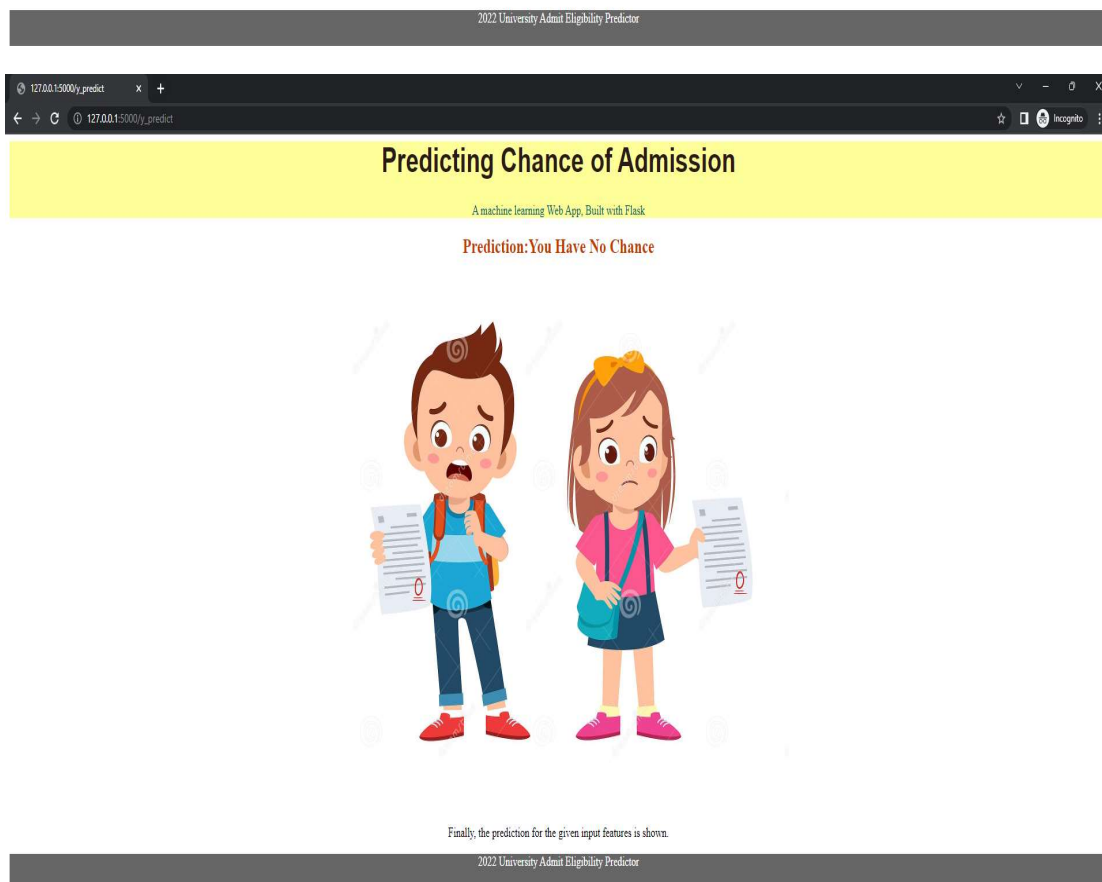
</div>
<p class="end-para">Finally, the prediction for the given input features is shown.</p>
</div>
<center>
<div class="footer">2022 University Admit Eligibility Predictor</div></center>
</body>
</html>

```

b. UI output screenshot



Finally, the prediction for the given input features is shown.



Finally, the prediction for the given input features is shown.