PROJECT REPORT

Project name: University Admit Eligibility Predictor Using IBM Watson

Table Of Contents:

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:

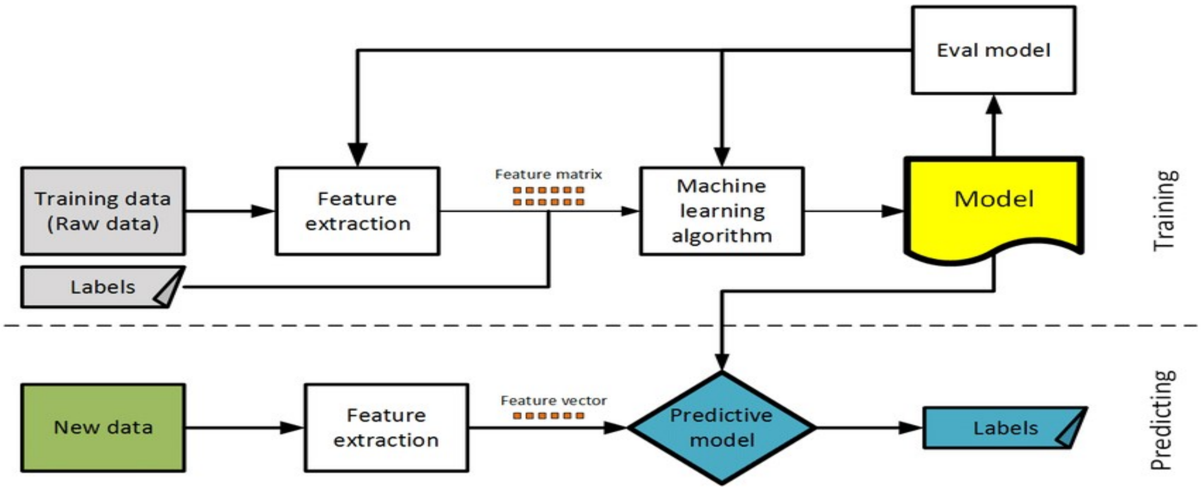
These problems can be resolved by using regression algorithms / classification algorithms as they can consider most of the features for prediction. Linear regression / KNN classification / Random forest regression can be used as the machine learning model for the model. XG boost model can also be used 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

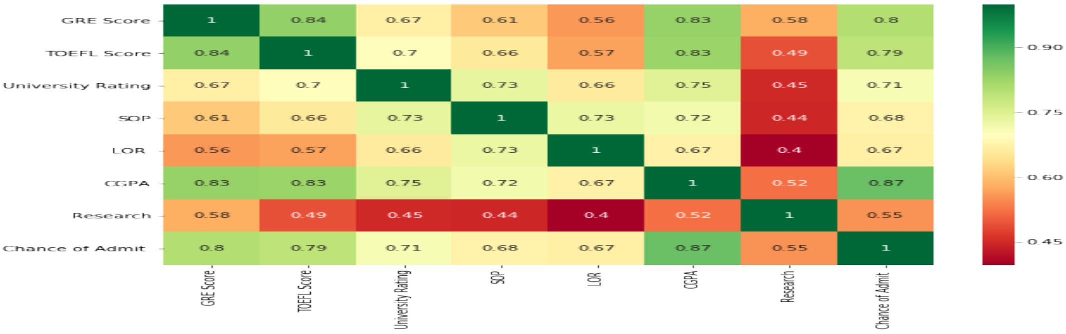
SOFTWARE DESIGNING:

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.

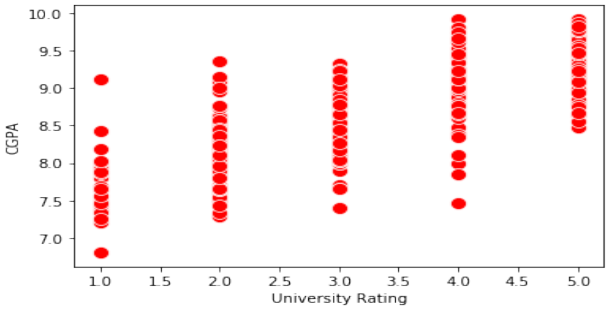
4. EXPERIMENTAL ANALYSIS

Heat map of the data



Pair plot

Scatter plot

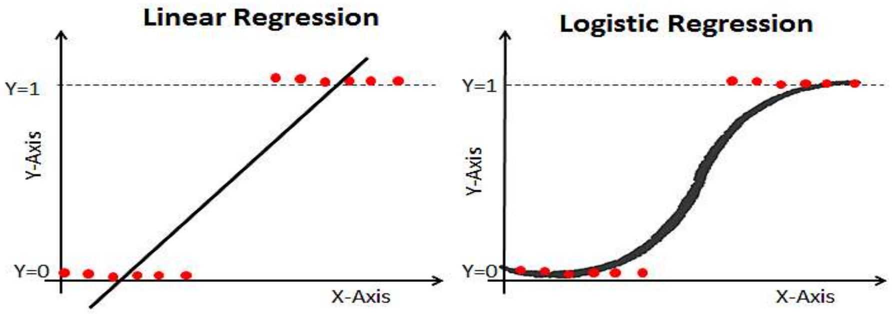


Logistic Regression:

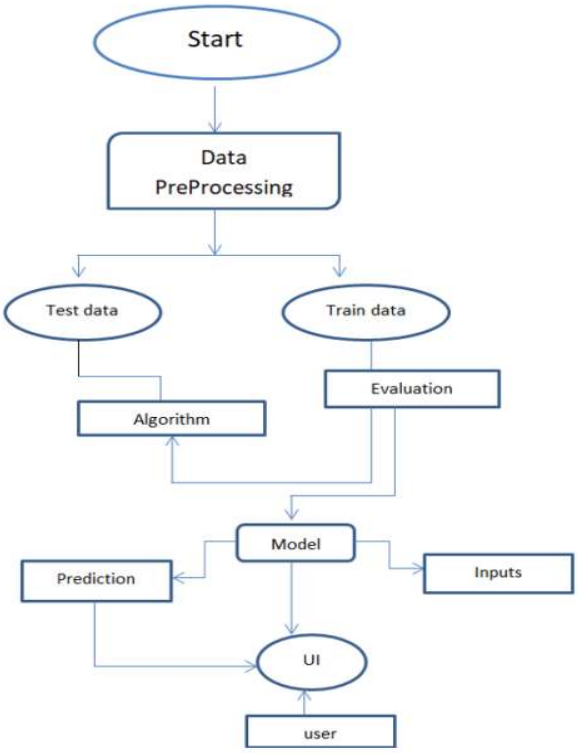
This type of statistical model(also known as *logit model*) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, 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.

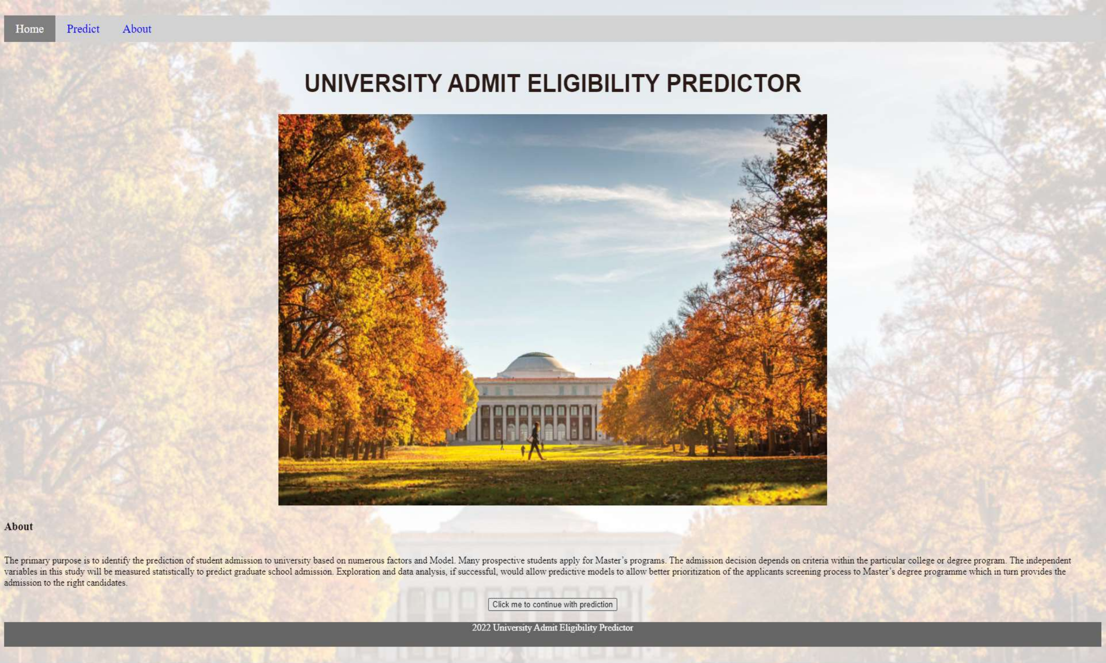


5. FLOWCHART



6. RESULT

Home page



UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

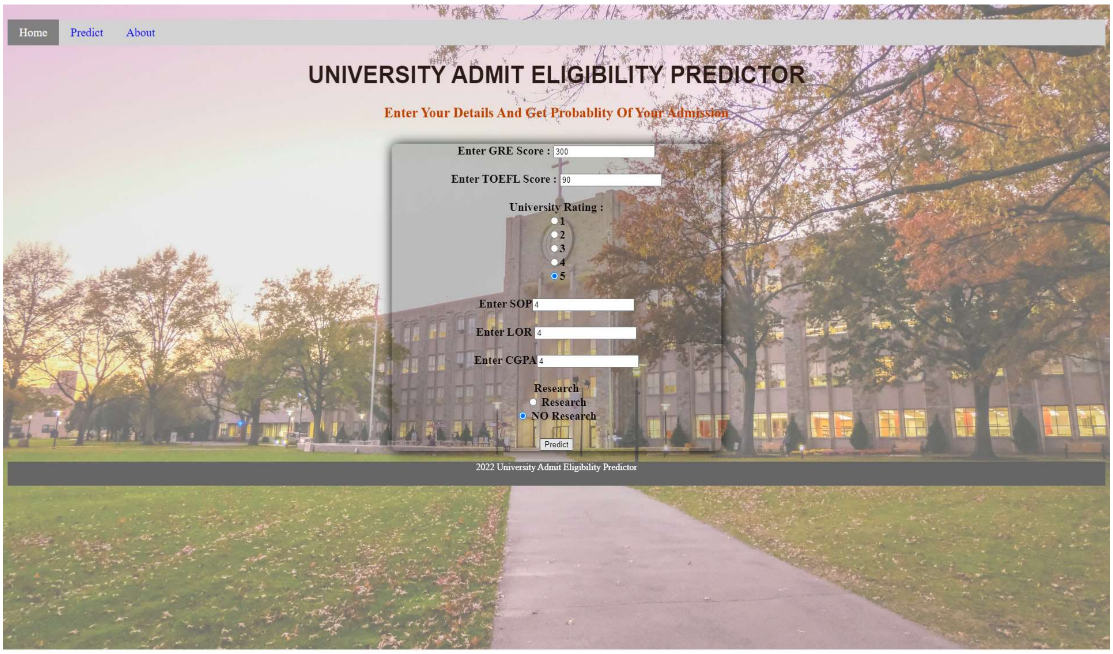
About

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.

2022 University Admit Eligibility Predictor

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:



UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

Enter Your Details And Get Probablity Of Your Admission

Enter GRE Score : 300

Enter TOEFL Score : 90

University Rating :
○ 1
○ 2
○ 3
○ 4
● 5

Enter SOP 4

Enter LOR 4

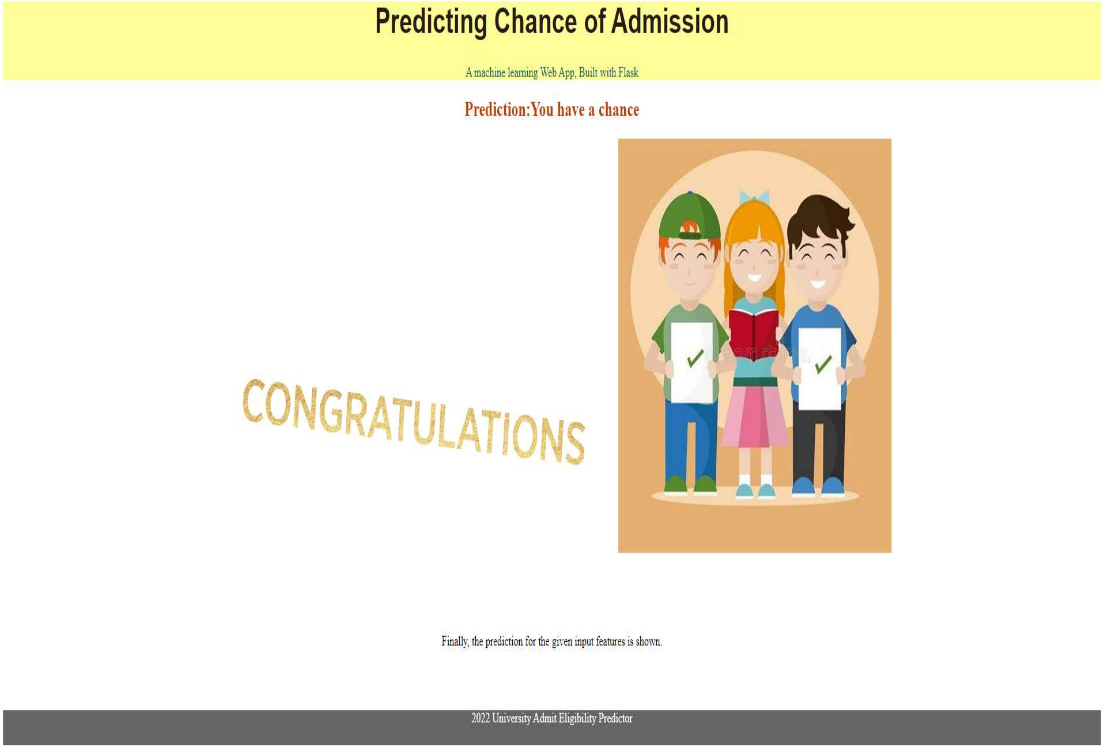
Enter CGPA 4

Research
○ Research
● NO Research

2022 University Admit Eligibility Predictor

Chance.html

→If student getting good results prediction shows your eligible.



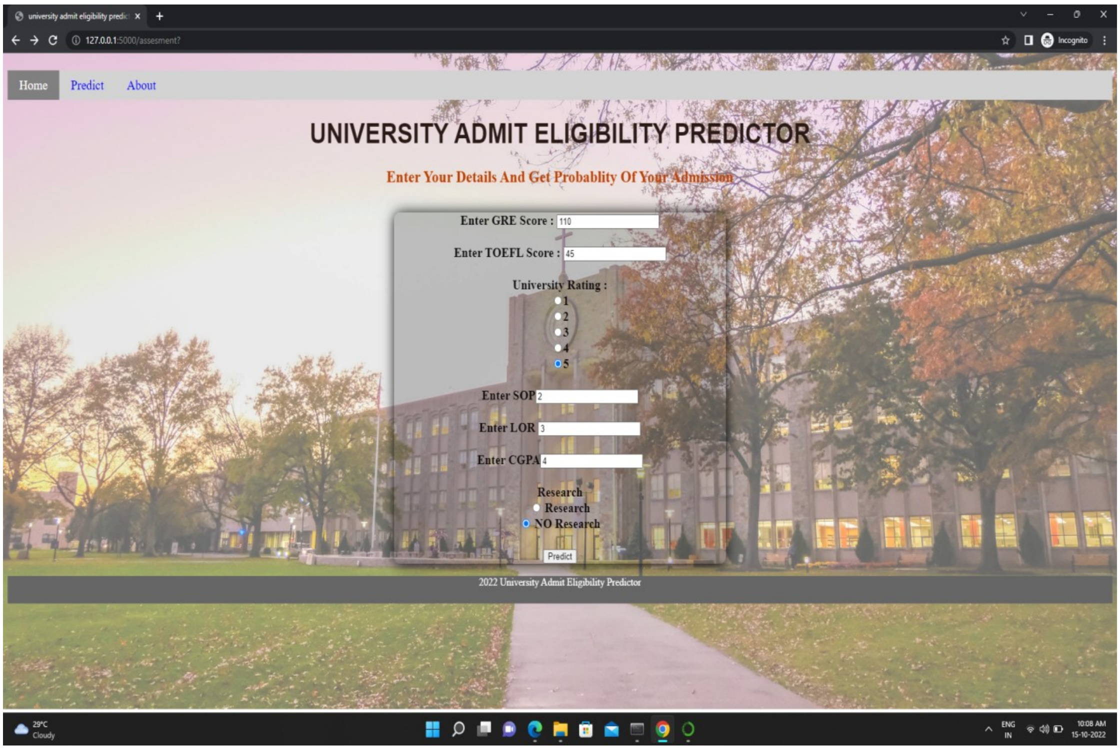
Predicting Chance of Admission

A machine learning Web App, Built with Flask

Prediction:You have a chance

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

2022 University Admit Eligibility Predictor



university admit eligibility predic
127.0.0.1:5000/assesment?
Incognito
Home
Predict
About

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

Enter Your Details And Get Probablity Of Your Admission

Enter GRE Score : 110

Enter TOEFL Score : 45

University Rating :
○ 1
○ 2
○ 3
○ 4
● 5

Enter SOP 2

Enter LOR 3

Enter CGPA 4

Research
○ Research
● NO Research

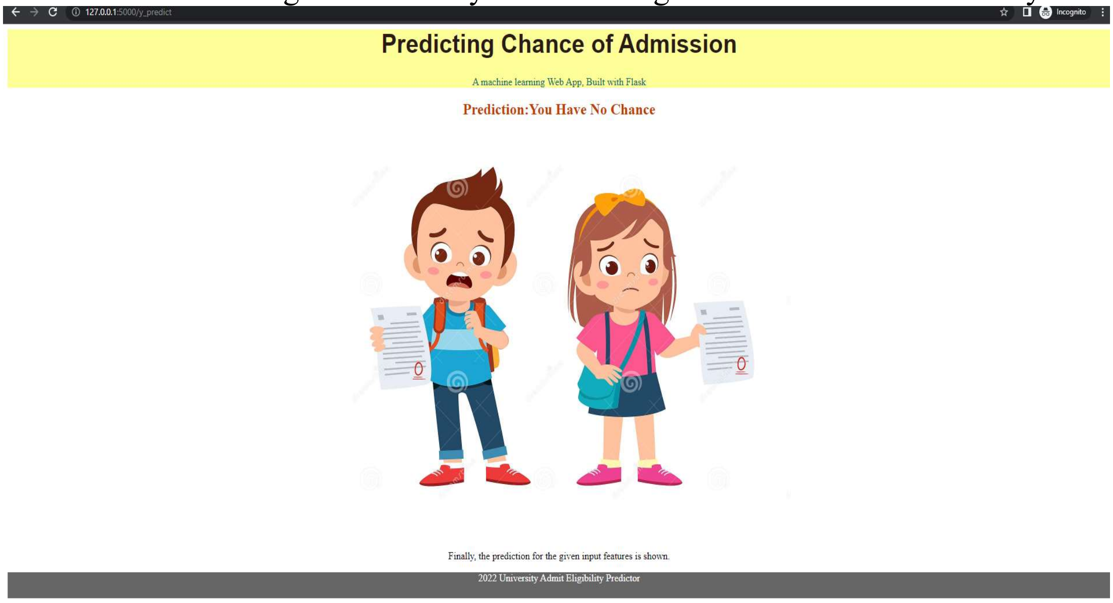
Predict

2022 University Admit Eligibility Predictor

29°C
Cloudy
ENG
IN
10:08 AM
15-10-2022

noChance.html

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


127.0.0.1:5000/y_predict

Predicting Chance of Admission

A machine learning Web App, Built with Flask

Prediction:You Have No Chance

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

2022 University Admit Eligibility Predictor

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. BIBILIOGRAPHY

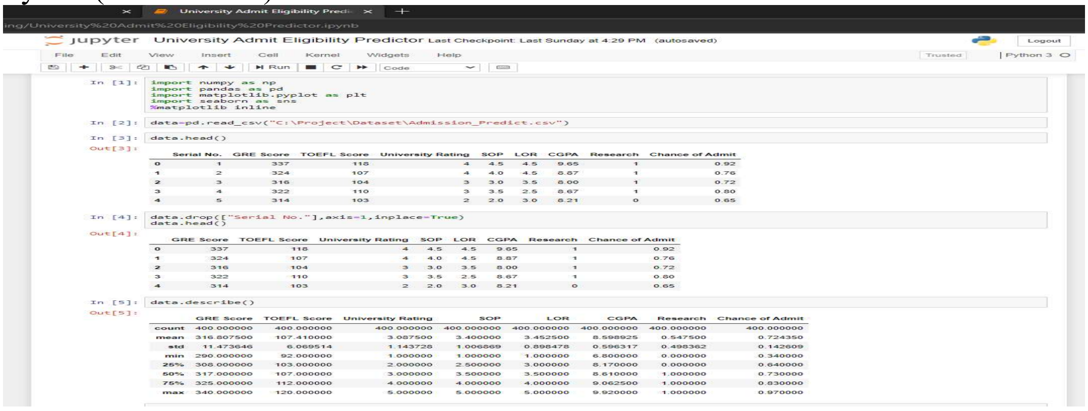
https://towardsdatascience.com/predicting-h-1b-status-using-random-forest-dc199a6d254c

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

APPENDIX

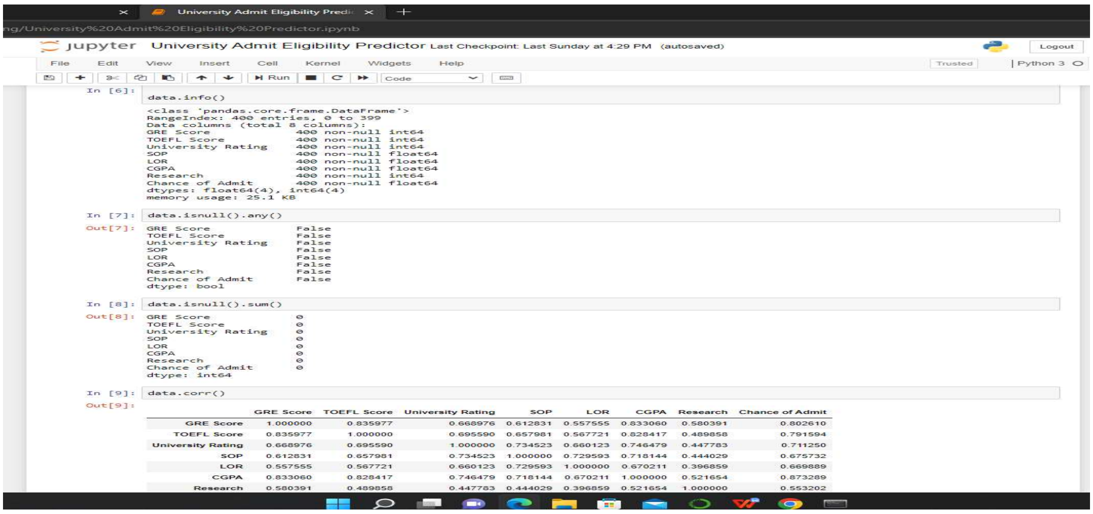
a. Source code

Python(source Code)



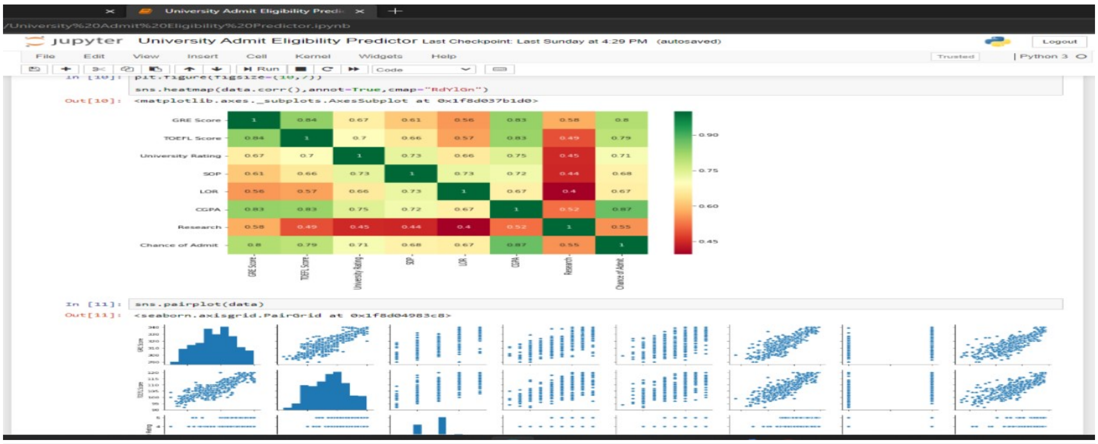
University Admit Eligibility Predi
ing/University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline

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          118                  4  4.5  4.5  9.65         1             0.92
1           2        324          107                  4  4.0  4.5  8.87         1             0.76
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

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          118                  4  4.5  4.5  9.65         1             0.92
1        324          107                  4  4.0  4.5  8.87         1             0.76
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

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   316.807500   107.410000           3.087500    3.400000    3.452500    8.598925    0.547500         0.724350
std     11.473646     6.069514           1.143728    1.006869    0.898478    0.596317    0.498362         0.142609
min    290.000000    92.000000           1.000000    1.000000    1.000000    6.800000    0.000000         0.340000
25%    308.000000   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    9.062500    1.000000         0.830000
max    340.000000   120.000000           5.000000    5.000000    5.000000    9.920000    1.000000         0.970000
```

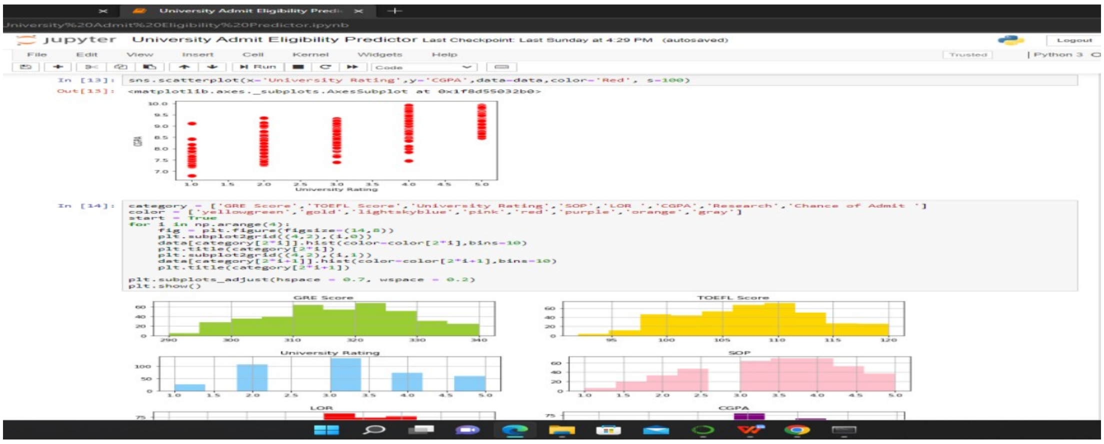


University Admit Eligibility Predi
ng/University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

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

        <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 int64
        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     0.835977           0.668976  0.612831  0.557555  0.833060  0.580391         0.802610
      TOEFL Score   0.835977     1.000000           0.695590  0.657981  0.567721  0.828417  0.489858         0.791594
University Rating   0.668976     0.695590           1.000000  0.734523  0.660123  0.746479  0.447783         0.711250
              SOP   0.612831     0.657981           0.734523  1.000000  0.729593  0.718144  0.444029         0.675732
              LOR   0.557555     0.567721           0.660123  0.729593  1.000000  0.670211  0.396859         0.669889
             CGPA   0.833060     0.828417           0.746479  0.718144  0.670211  1.000000  0.521654         0.873289
         Research   0.580391     0.489858           0.447783  0.444029  0.396859  0.521654  1.000000         0.553202
```



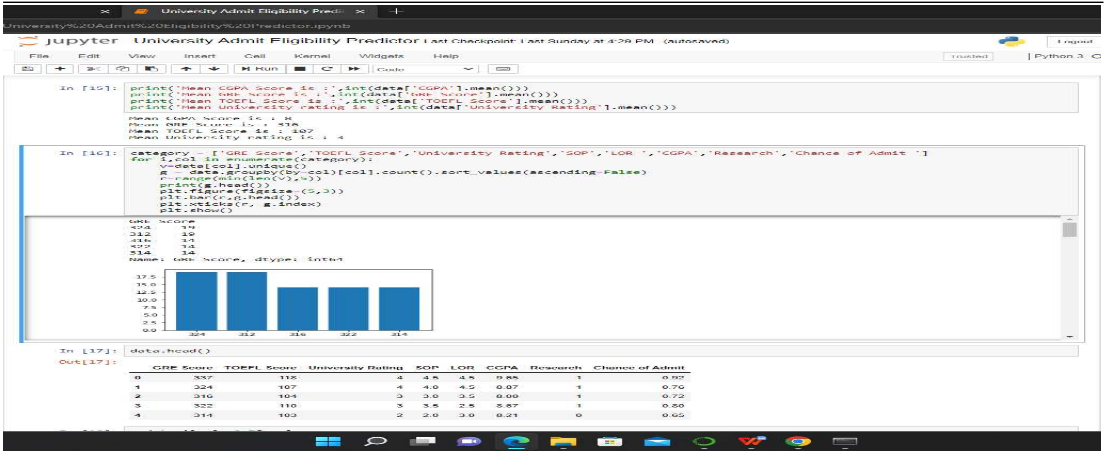
```
In [10]: plt.figure(figsize=(10,7))
         sns.heatmap(data.corr(),annot=True,cmap="RdYlGn")
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1f8d037b1d0>

In [11]: sns.pairplot(data)
Out[11]: <seaborn.axisgrid.PairGrid at 0x1f8d04983c8>
```


University Admit Eligibility Predi
/University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [13]: sns.scatterplot(x='University Rating',y='CGPA',data=data,color='Red', s=100)
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1f8d55032b0>
```

10.0
9.5
9.0
8.5
8.0
7.5
7.0
CGPA
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
University Rating

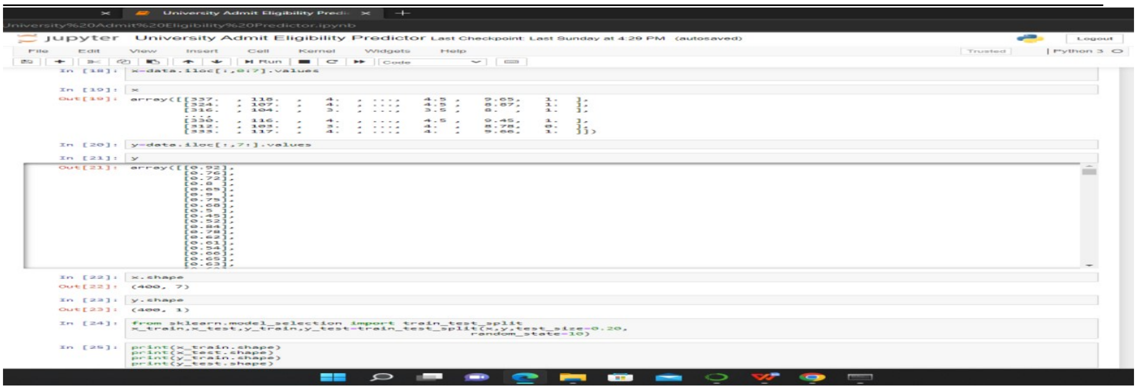
```
In [14]: category = ['GRE Score','TOEFL Score','University Rating','SOP','LOR ','CGPA','Research','Chance of Admit ']
         color = ['yellowgreen','gold','lightskyblue','pink','red','purple','orange','gray']
         start = True
         for i in np.arange(4):
             fig = plt.figure(figsize=(14,8))
             plt.subplot2grid((4,2),(i,0))
             data[category[2*i]].hist(color=color[2*i],bins=10)
             plt.title(category[2*i])
             plt.subplot2grid((4,2),(i,1))
             data[category[2*i+1]].hist(color=color[2*i+1],bins=10)
             plt.title(category[2*i+1])

         plt.subplots_adjust(hspace = 0.7, wspace = 0.2)
         plt.show()
```


University Admit Eligibility Predi
University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

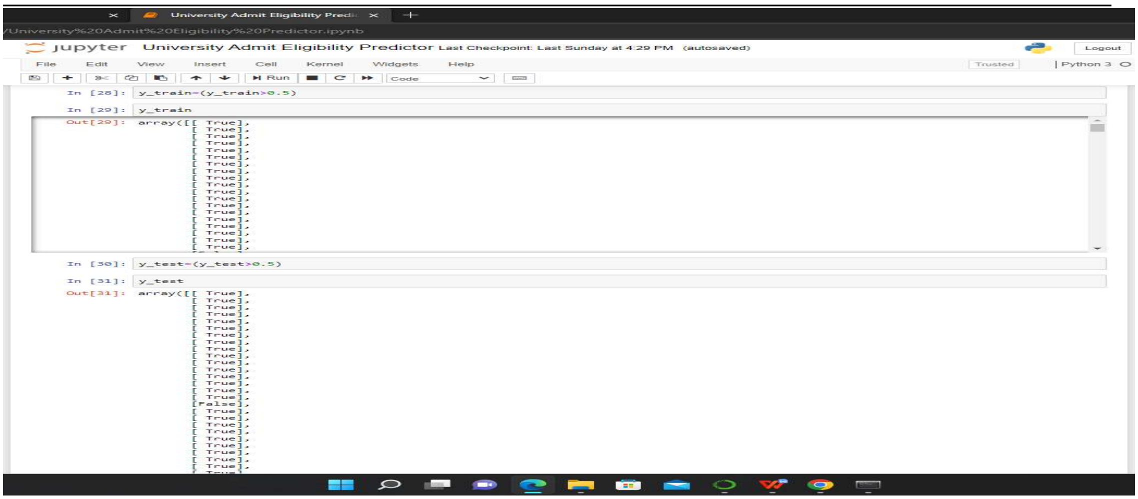
```
In [15]: print('Mean CGPA Score is :',int(data['CGPA'].mean()))
         print('Mean GRE Score is :',int(data['GRE Score'].mean()))
         print('Mean TOEFL Score is :',int(data['TOEFL Score'].mean()))
         print('Mean University rating is :',int(data['University Rating'].mean()))

         Mean CGPA Score is : 8
         Mean GRE Score is : 316
         Mean TOEFL Score is : 107
         Mean University rating is : 3

In [16]: category = ['GRE Score','TOEFL Score','University Rating','SOP','LOR ','CGPA','Research','Chance of Admit ']
         for i,col in enumerate(category):
             v=data[col].unique()
             g = data.groupby(by=col)[col].count().sort_values(ascending=False)
             r=range(min(len(v),5))
             print(g.head())
             plt.figure(figsize=(5,3))
             plt.bar(r,g.head())
             plt.xticks(r, g.index)
             plt.show()

         GRE Score
         324    19
         312    19
         316    14
         322    14
         314    14
         Name: GRE Score, dtype: int64
```

```
In [17]: data.head()
Out[17]:
```

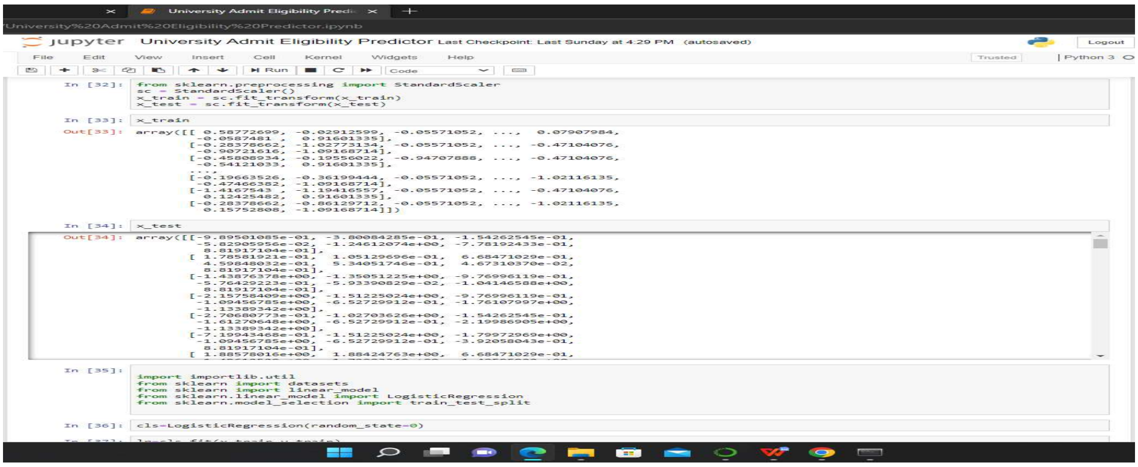
	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	337	118	4	4.5	4.5	9.65	1	0.92
1	324	107	4	4.0	4.5	8.87	1	0.76
2	316	104	3	3.0	3.5	8.00	1	0.72
3	322	110	3	3.5	2.5	8.67	1	0.80
4	314	103	2	2.0	3.0	8.21	0	0.65


University Admit Eligibility Predi
University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [18]: x=data.iloc[:,0:7].values

In [19]: x
Out[19]: array([[337.  , 118.  ,   4.  , ...,   4.5 ,   9.65,   1.  ],
                [324.  , 107.  ,   4.  , ...,   4.5 ,   8.87,   1.  ],
                [316.  , 104.  ,   3.  , ...,   3.5 ,   8.  ,   1.  ],
                ...,
                [330.  , 116.  ,   4.  , ...,   4.5 ,   9.45,   1.  ],
                [312.  , 103.  ,   3.  , ...,   4.  ,   8.78,   0.  ],
                [333.  , 117.  ,   4.  , ...,   4.  ,   9.66,   1.  ]])

In [20]: y=data.iloc[:,7:].values

In [21]: y
Out[21]: array([[0.92],
                [0.76],
                [0.72],
                [0.8 ],
                [0.65],
                [0.9 ],
                [0.75],
                [0.68],
                [0.5 ],
                [0.45],
                [0.52],
                [0.84],
                [0.78],
                [0.62],
                [0.61],
                [0.54],
                [0.66],
                [0.65],
                [0.63],

In [22]: x.shape
Out[22]: (400, 7)

In [23]: y.shape
Out[23]: (400, 1)

In [24]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,
                                                        random_state=10)

In [25]: print(x_train.shape)
         print(x_test.shape)
         print(y_train.shape)
         print(y_test.shape)
```

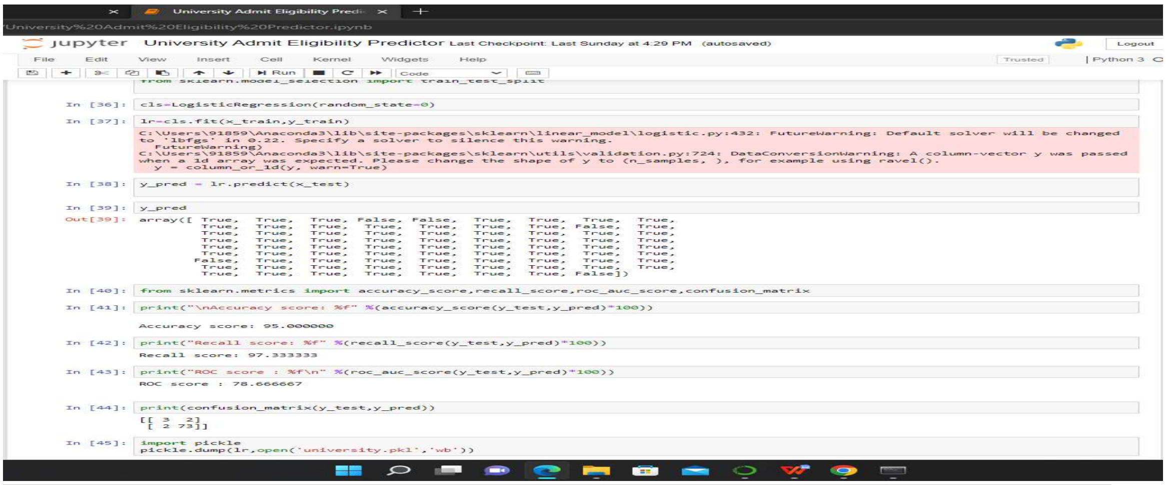

University Admit Eligibility Predi
/University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
In [28]: y_train=(y_train>0.5)

In [29]: y_train
Out[29]: array([[ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],

In [30]: y_test=(y_test>0.5)

In [31]: y_test
Out[31]: array([[ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [False],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
                [ True],
```



University Admit Eligibility Predi
/University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [32]: from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         x_train = sc.fit_transform(x_train)
         x_test = sc.fit_transform(x_test)

In [33]: x_train
Out[33]: array([[ 0.58772699, -0.02912599, -0.05571052, ...,  0.07907984,
                 -0.0587481 ,  0.91601335],
                [-0.28378662, -1.02773134, -0.05571052, ..., -0.47104076,
                 -0.90721616, -1.09168714],
                [-0.45808934, -0.19556022, -0.94707888, ..., -0.47104076,
                 -0.54121033,  0.91601335],
                ...,
                [-0.19663526, -0.36199444, -0.05571052, ..., -1.02116135,
                 -0.47466382, -1.09168714],
                [-1.4167543 , -1.19416557, -0.05571052, ..., -0.47104076,
                  0.12425482,  0.91601335],
                [-0.28378662, -0.86129712, -0.05571052, ..., -1.02116135,
                  0.15752808, -1.09168714]])

In [34]: x_test
Out[34]: array([[-9.89501085e-01, -3.80084285e-01, -1.54262545e-01,
                 -5.82905956e-02, -1.24612074e+00, -7.78192433e-01,
                  8.81917104e-01],
                [ 1.78581921e-01,  1.05129696e-01,  6.68471029e-01,
                  4.59848032e-01,  5.34051746e-01,  4.67310370e-02,
                  8.81917104e-01],
                [-1.43876378e+00, -1.35051225e+00, -9.76996119e-01,
                 -5.76429223e-01, -5.93390829e-02, -1.04146588e+00,
                  8.81917104e-01],
                [-2.15758409e+00, -1.51225024e+00, -9.76996119e-01,
                 -1.09456785e+00, -6.52729912e-01, -1.76107997e+00,
                 -1.13389342e+00],
                [-2.70680773e-01, -1.02703626e+00, -1.54262545e-01,
                 -1.61270648e+00, -6.52729912e-01, -2.19986905e+00,
                 -1.13389342e+00],
                [-7.19943468e-01, -1.51225024e+00, -1.79972969e+00,
                 -1.09456785e+00, -6.52729912e-01, -3.92058043e-01,
                  8.81917104e-01],
                [ 1.88578016e+00,  1.88424763e+00,  6.68471029e-01,

In [35]:
         import importlib.util
         from sklearn import datasets
         from sklearn import linear_model
         from sklearn.linear_model import LogisticRegression
         from sklearn.model_selection import train_test_split

In [36]: cls=LogisticRegression(random_state=0)
```



University Admit Eligibility Predi
/University%20Admit%20Eligibility%20Predictor.ipynb
jupyter University Admit Eligibility Predictor Last Checkpoint: Last Sunday at 4:29 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
         from sklearn.model_selection import train_test_split

In [36]: cls=LogisticRegression(random_state=0)

In [37]: lr=cls.fit(x_train,y_train)
         C:\Users\91859\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
         to 'lbfgs' in 0.22. Specify a solver to silence this warning.
           FutureWarning)
         C:\Users\91859\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed
         when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
           y = column_or_1d(y, warn=True)

In [38]: y_pred = lr.predict(x_test)

In [39]: y_pred
Out[39]: array([ True,  True,  True, False, False,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True, False,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                False,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True,  True,  True,
                 True,  True,  True,  True,  True,  True,  True, False])

In [40]: from sklearn.metrics import accuracy_score,recall_score,roc_auc_score,confusion_matrix

In [41]: print("\nAccuracy score: %f" %(accuracy_score(y_test,y_pred)*100))

         Accuracy score: 95.000000

In [42]: print("Recall score: %f" %(recall_score(y_test,y_pred)*100))
         Recall score: 97.333333

In [43]: print("ROC score : %f\n" %(roc_auc_score(y_test,y_pred)*100))
         ROC score : 78.666667

In [44]: print(confusion_matrix(y_test,y_pred))
         [[ 3  2]
          [ 2 73]]

In [45]: import pickle
         pickle.dump(lr,open('university.pkl','wb'))
```

App.py

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
app = Flask(__name__,template_folder="Template")
model = pickle.load(open("university.pkl", "rb"))
@app.route('/')
def index():
    return render_template("index.html")
@app.route('/home1')
def home1():
    return render_template("index.html")
@app.route("/assesment")
def home():
    return render_template('Demo2.html')
@app.route("/about")
def about():
    return render_template('about.html')
@app.route("/y_predict", methods = ["POST","GET"])
def y_predict():
    min1 = [290.0, 92.0, 1.0, 1.0, 1.0, 6.8, 0.0]
    max1 = [340.0, 120.0, 5.0, 5.0, 5.0, 9.92, 1.0]
    k = [float(x) for x in request.form.values()]
    print(k)
    p = []
    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>
<img src="../static/image/20131114DD078-1-864x576.jpg" width="50%" height="50%"  >
</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>
<lable for="University Rating"> University Rating : </lable><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

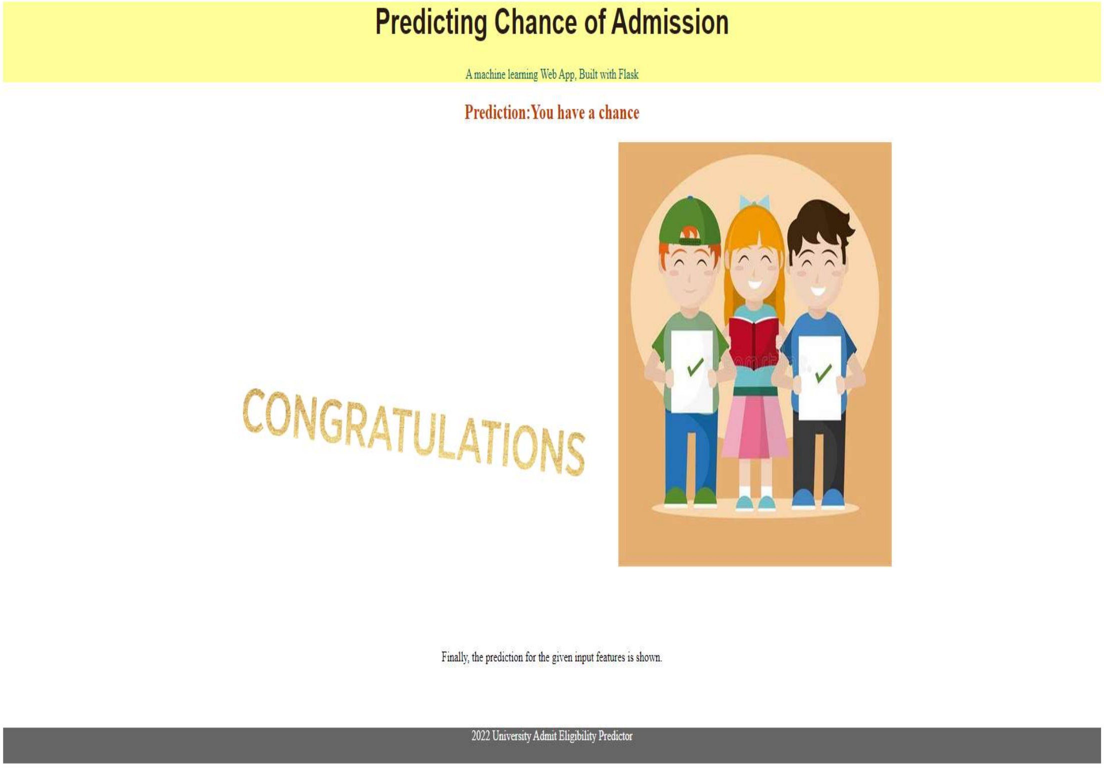
```
<html>
```

```
<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></h1>
<img src="../static/image/source.gif"/>
<img src="../static/image/th (5).jpg"/>
</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></h1>
<img src="../static/image/nochance%20(1).png"/>
</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



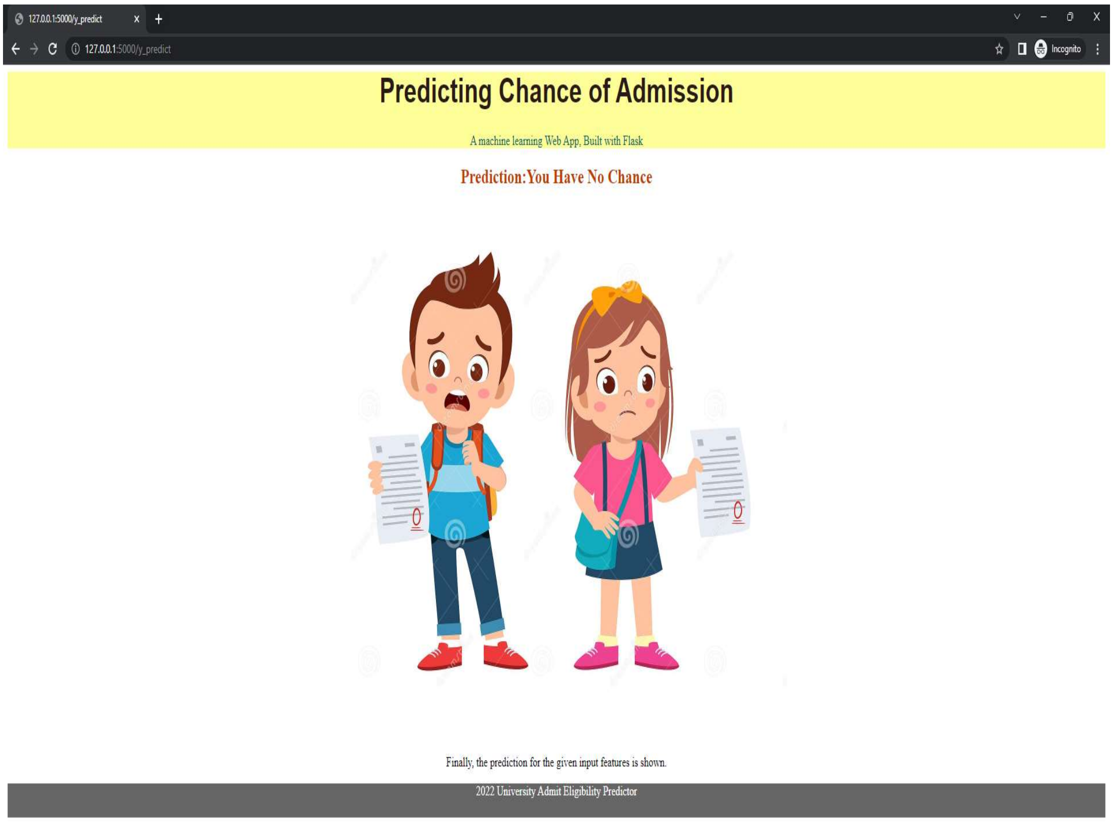
Predicting Chance of Admission

A machine learning Web App, Built with Flask

Prediction:You have a chance

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

2022 University Admit Eligibility Predictor



Predicting Chance of Admission

A machine learning Web App, Built with Flask

Prediction:You Have No Chance

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

2022 University Admit Eligibility Predictor