GRADUATE ADMISSION PREDICTION

MINOR PROJECT

FINAL REPORT

Submitted By:

Sanyam Jain (9919103094)

Akshit Tyagi (9919103101)

Geet Jindal (9919103105)

Mudit Virmani (9919103106)

UNDER SUPERVISION OF:

SHARIQ MURTUZA

DECEMBER 2021

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Mr. SHARIQ MURTUZA, Jaypee Institute of Information Technology, India for his/her generous guidance, help and useful suggestions.

I also wish to extend my thanks to project partners and other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signature(s) of Students

Sanyam Jain (9919103094)

Akshit Tyagi (9919103101)

Geet Jindal (9919103105)

Mudit Virmani (9919103106)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place:

Date: December 1,2021

Name:	**Enrolment:**
Sanyam Jain	9919103094
Akshit Tyagi	9919103101
Geet Jindal	9919103105
Mudit Virmani	9919103106

CERTIFICATE

This is to certify that the work titled “Graduate Admission Prediction” submitted by Name of Students of B.Tech.(CSE) of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Digital Signature of Supervisor

Name of Supervisor: Mr. Shariq Murtuza

Designation: ASSISTANT PROFESSOR(GRADE-II)

Date: December 1,2021

ABSTRACT

Students seek help from many sources such as online sites or professional professionals to find the best options for their future. A good career counsellor charges a lot of money for providing such solutions. Online sources are also unreliable as information from certain sources is not always accurate. Students also do their own analysis before applying to any of the institutions, but this approach is slow and certainly not consistent with getting real results and may even involve human error. The purpose of this project is to predict the admission of a student based on different features

INTRODUCTION

International academic test can be an easy way to predict how a university / college person will work and be impartial and completely transparent. Individuals will no longer need to rely on consultative institutions that may deviate slightly from the list of colleges / universities that may be in agreement with them. In addition, applying only to colleges / universities where a student has a real opportunity can slow down the application process. In addition, the cost of living in the area where the college / university is located can also be provided on the website.

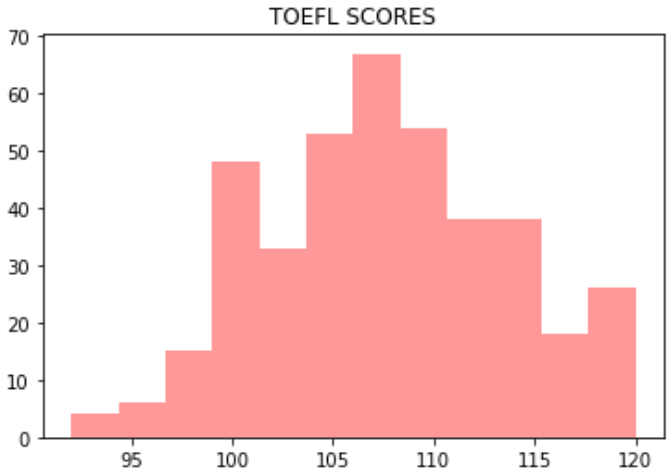
The main purpose of this project is to help students save their time and money to spend on educational consultation costs. It will also help them reduce their application costs to a minimum by showing them a proposal for universities where they have a better chance of gaining admission thus saving a lot of money on application fees.

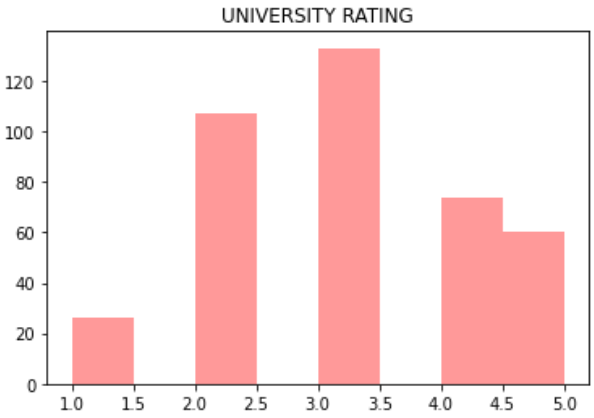
BACKGROUND STUDY

Linear Regression is the most basic algorithm in Machine Learning. It is a regression algorithm which means that it is useful when we are required to predict continuous values, that is, the output variable 'y' is continuous in nature.
Linear regression assumes linear relation between x and y. The hypothesis function for linear regression is y = m1.x1 + m2.x2 + m3.x3 + … + mn.xn + b where m1 , m2 , m3 are called the parameters and b is the intercept of the line. This equation shows that the output variable y is linearly dependent on the features x1 , x2 , x3. The more you are dependent on a particular feature, more will be the value of corresponding m for that feature. We can find out which feature is more important or which feature is more affecting the result by varying the values of m one at a time and see if it is affecting the result, that is , the value of y. So, here in order to predict the values of y for given features values (x values) we use this equation. But what we are missing here is the values of parameters (m1 , m2 , m3 , … and b). So, we will be using our training data (where the values of x and y are already given) to find out values of parameters and later on predict the value of y for a set of new values of x.

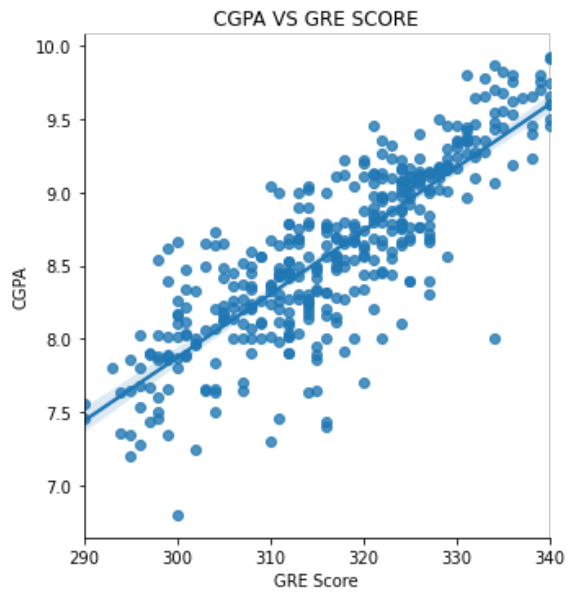
Graphs:

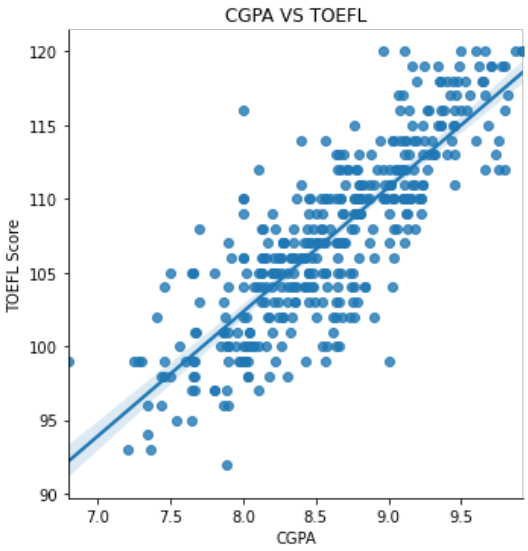
Distplot: A Distplot or distribution plot, depicts the variation in the data distribution. The Distplot depicts the data by a histogram and a line in combination to it.



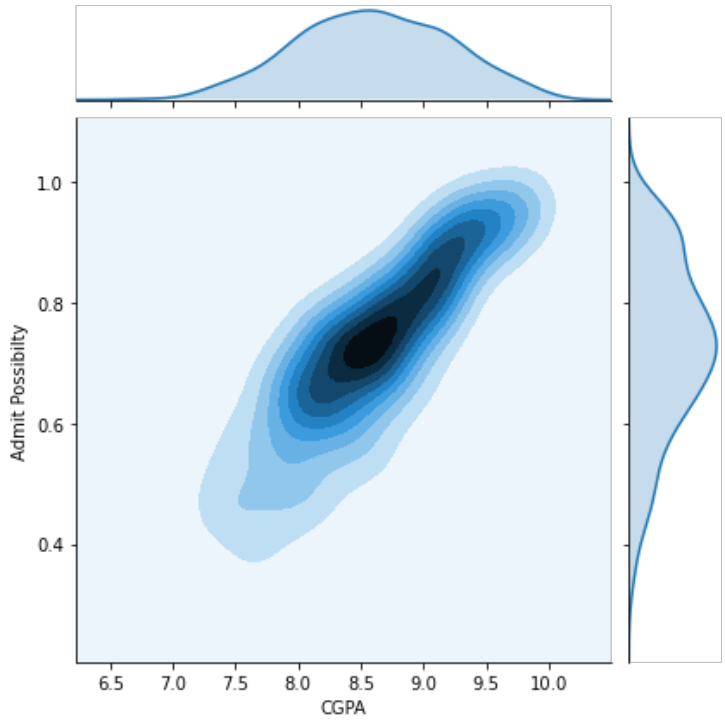


Lmplot: is used to draw a scatter plot onto a FacetGrid

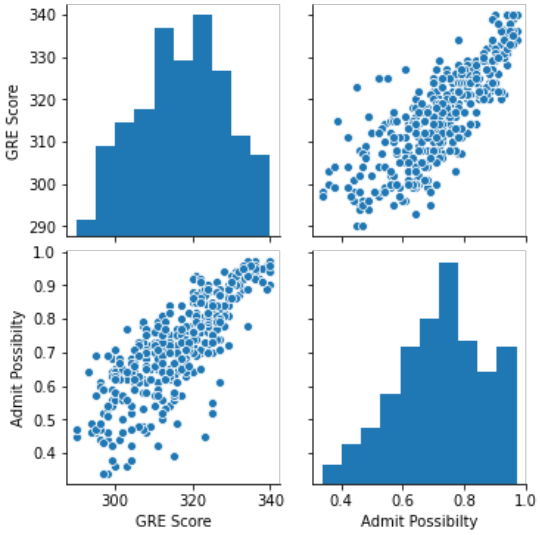




Jointplot: A Jointplot comprises three plots. Out of the three, one plot displays a bivariate graph which shows how the dependent variable(Y) varies with the independent variable(X). Another plot is placed horizontally at the top of the bivariate graph and it shows the distribution of the independent variable(X).



Pairplot: A pairplot plot a pairwise relationships in a dataset. The pairplot function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column.



REQUIREMENT ANALYSIS

Anaconda: is a distribution of the Python, The distribution includes data-science packages suitable for Windows, Linux, and macOS.

Jupyter Notebook: is an excellent open-source web application that allows you to create and share documents that contain live code, equations, visualizations and used for data cleaning and transformation, numerical simulation, statistical modelling, data visualization and machine learning.

Python Libraries for Exploratory Data Analysis: NumPy, Pandas, Matplotlib and Seaborn.

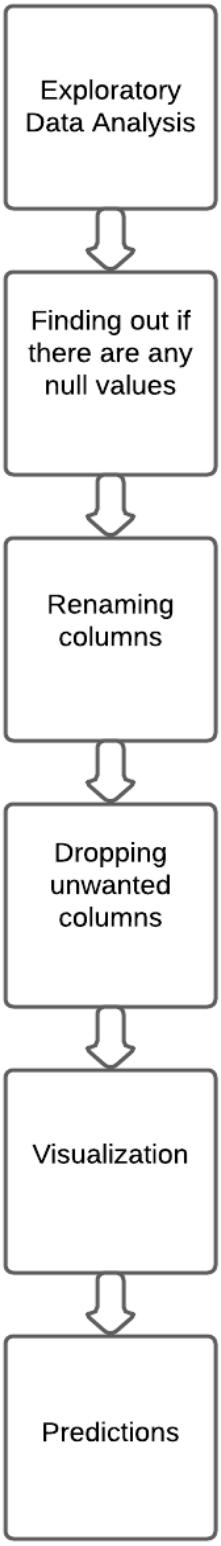
Python Libraries for machine learning: Sklearn is an excellent source for implementing machine learning algorithms.

Flask: is an API of Python that allows us to build up web-applications.

Hardware requirements:

- 32- or 64-bit computer.
- Minimum 3 GB disk space to download and install anaconda.

DETAILED DESIGN



IMPLEMENTATION

Prediction of Probability of You Getting an Admit in the Foreign Universities

The dataset contains several arguments(basically the inputs you need to provide) which are considered for the application for University. The parameters included are :

GRE Scores (out of 340) TOEFL Scores (out of 120) Under Grad University Rating (out of 5) Statement of Purpose (out of 5) Letter of Recommendation Strength (out of 5) Undergraduate GPA (out of 10) Research Experience (either 0 or 1) Chance of Admit (ranging from 0 to 1)

Dataset: https://www.kaggle.com/mohansacharya/graduate-admissions/home

Importing the necessary libraries for implementing Analysis.

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

Exploratory Data Analysis

As the Dataset is clean and has no null values we are directly going to implement the Exploration

In [2]:
```
Reading = pd.read_csv("datasets_14872_228180_Admission_Predict.csv")
Reading.head() #printing the first five rows
```

Out[2]:

	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 [3]:
```
Reading.describe()
```

Out[3]:

	Serial No.	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	400.000000
mean	200.500000	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	115.614301	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	1.000000	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	100.750000	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	200.500000	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	300.250000	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	400.000000	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

Finding out if there are any null values

In [4]:
```
Null=Reading.isnull()
Null.sum()
```

Out[4]:
```
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
```

Renaming columns

In [5]:
```
Reading = Reading.rename(columns={'GRE Score': 'GRE Score', 'TOEFL Score': 'TOEFL Score', 'LOR ': 'LOR', 'Chance of Admit ': 'Adm
Reading.head()
```

Out[5]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Admit Possibilty
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

Dropping unwanted columns

In [6]:
```
Reading.drop('Serial No.', axis='columns', inplace=True)
Reading.head()
```

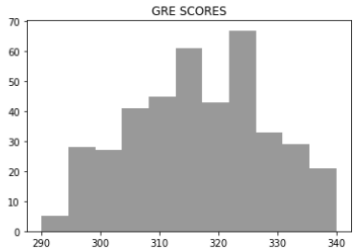
Out[6]:

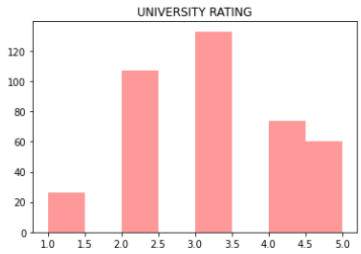
	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Admit Possibilty
0	337	118	4	4.5	4.5	9.65	1	0.92
1	324	107	4	4.0	4.5	8.87	1	0.76
2	316	104	3	3.0	3.5	8.00	1	0.72
3	322	110	3	3.5	2.5	8.67	1	0.80
4	314	103	2	2.0	3.0	8.21	0	0.65

Visualization

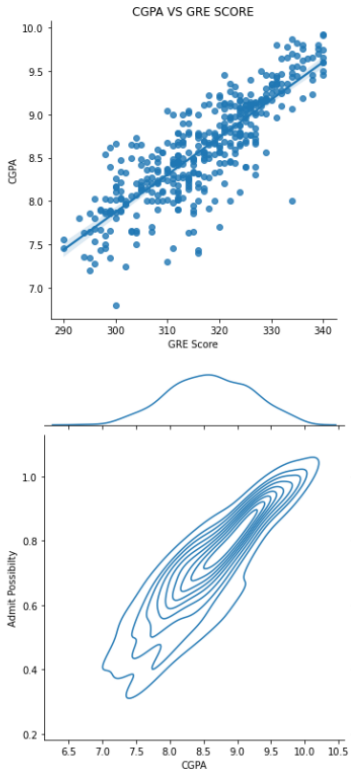
In [7]:
```
gre_score = Reading[["GRE Score"]] #selecting only the required coloumn
toefl_score = Reading[["TOEFL Score"]]
uni_rating=  Reading[["University Rating"]]
```

In [8]:
```
fig=sns.distplot(gre_score,color='black',kde=False)
plt.title("GRE SCORES")
plt.show()

fig=sns.distplot(toefl_score,color='r',kde=False)
plt.title("TOEFL SCORES")
plt.show()

fig=sns.distplot(uni_rating,color='r',kde=False)
plt.title("UNIVERSITY RATING")
plt.show()
```



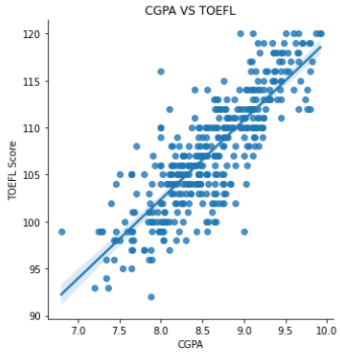


In [9]:
```
fig=sns.lmplot(x='GRE Score',y='CGPA',data=Reading)
plt.title("CGPA VS GRE SCORE")
plt.show()


fig=sns.jointplot(x='CGPA',y='Admit Possibilty',data=Reading,kind='kde')
plt.show()
```

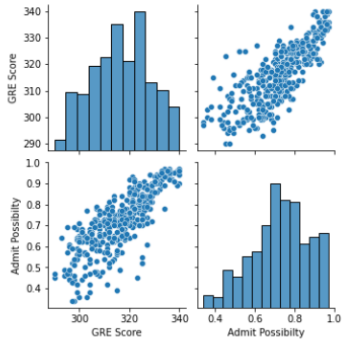


From the above plot we can see that the person who did well in UG also did well in GRE.

In [10]:
```
fig=sns.lmplot(x='CGPA',y='TOEFL Score',data=Reading)
plt.title("CGPA VS TOEFL")
plt.show()
```



In [11]:
```
sns.pairplot(data=Reading,vars=["GRE Score","Admit Possibilty"])
plt.show()
```



We can see that the GRE SCORE is a deal breaker for getting an admit

Predictions

We are using a linear regression model. Why?

This is a supervised model data and also the independent variable X having the parameters GRE, TOEFL etc are in high relationship with the dependent varaible y being the chance of admit.

In [12]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
```

Splitting the data as x and y where x contains the dependent varaiable data and y contains the independent variable data

In [13]:
```
x=Reading.drop('Admit Possibilty',axis='columns')
y=Reading['Admit Possibilty']
x_train,x_test,y_train,y_test=train_test_split(x, y)
```

Further Splitting the data as test and train where the train set contains the 80% of data and the test set contains 20% of the data where we can see that 300 out of 400 rows are taken

In [14]:
```
x_train.shape
```

Out[14]: (300, 7)

In [15]:
```
x_test.shape
```

Out[15]: (100, 7)

In [16]:
```
y_train.shape
```

Out[16]: (300,)

In [17]:
```
y_test.shape
```

Out[17]: (100,)

In [18]:
```
linear_regression = LinearRegression()
linear_regression = linear_regression.fit(x_train,y_train)
```

A test set should still be held out for final evaluation, but the validation set is no longer needed when doing CV. In the basic approach, called k-fold CV, the training set is split into k smaller sets

In [19]:
```
def get_cv_scores(linear_regression):
    scores = cross_val_score(linear_regression,
                             x_train,
                             y_train,
                             cv=5,
                             scoring='r2')

    print('CV Mean: ', np.mean(scores))
    print('STD: ', np.std(scores))
    print('\n')

# get cross val scores
get_cv_scores(linear_regression)
```

```
CV Mean:  0.7774427116035557
STD:  0.05667781205946649
```

The CV score says that the model is neither an underfit nor an overfit. Any value between 0 and 1 is good.

The model is predicting on the test set

In [20]:
```
model = LinearRegression(normalize=True)
model.fit(x_test, y_test)
model.score(x_test, y_test)
```

Out[20]: 0.8239655998469342

Accuracy on the test set is 82.3%

The model finally predicts the data based on user input

In [21]:
```
print('The chance of you getting an admit in the US is {}%'.format(round(model.predict([[305, 108, 4, 4.5, 4.5, 8.35, 0]])[0]*10
```

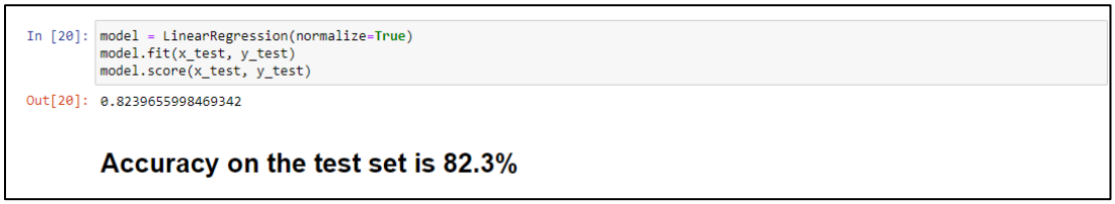
```
The chance of you getting an admit in the US is 71.1%
```

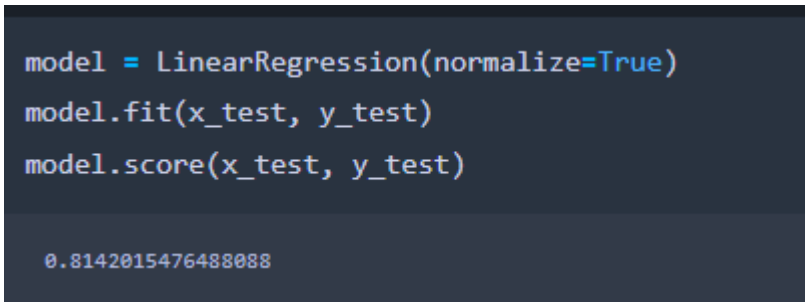
EXPERIMENTAL RESULT

DATASET USED: https://www.kaggle.com/mohansacharya/graduate-admissions/home

SHAPE OF DATASET: (400, 9)

Accuracy Analysis: On running the code on different test cases, below is the accuracy.





Algorithm used in this project varies +- 5% in accuracy as per acquired results.

CONCLUSION

Thus, with the help of Supervised Machine Learning and Exploratory Data Analysis, the prediction of the possibility of a candidate getting an admit has been successfully implemented

FUTURE SCOPE

From the proposed work we can identify only chance to get seat and we are not able to identify which university we are obtaining. So, in future we can develop a representation, which gives us a list of universities in which we can obtain admission.

REFERENCES

- https://jupyter-notebook.readthedocs.io/en/stable/
- https://pandas.pydata.org/docs/
- https://scikit-learn.org/0.21/documentation.html
- https://flask.palletsprojects.com/en/2.0.x/
- https://numpy.org/doc/
- https://seaborn.pydata.org/
- https://matplotlib.org/3.4.3/contents.html
- https://www.kaggle.com/mohansacharya/graduate-admissions/home