PROJECT REPORT

GRADUATE ADMISSION PREDICTION

SUBMITTED BY:- Vaishnavi Wasre
USN:- 3GN18CS119
DEPT:- COMPUTER SCIENCE

UNDERTAKING

I declare that the work presented in this project titled "UNIVERSITY ADMISSION PREDICTION", submitted to the TIMTS, for the award of the Internship in Data Science, is my original work. I have not plagiarized or submitted the same work for the award of any other Internship. In case this undertaking is found incorrect, I accept that my Project may be unconditionally withdrawn.

October,2021 Vaishnavi Wasre

CERTIFICATE

This is to certify that the work contained in the project titled “UNIVERSITY ADMISSION PREDICTION”, by Vaishnavi Wasre, has been carried out under my supervision and that this work has not been submitted elsewhere for internship.

Times Institute of Management and Technical Studies

Data science and ML
Bengaluru, Karnataka 560103

ACKNOWLEDGEMENT

I take upon this opportunity to acknowledge the many people whose prayers and support meant a lot to me. I am deeply indebted to my mentor **Mr. Sumit Chatterjee** who motivated me along the way.

I would like to thank all my teachers who help me in this project.
I further thank my friends.My heartfelt thanks to parents who supported me a lot. I owe my sincere gratitude towards the almighty God.

Finally, I would like to wind up by paying my heartfelt thanks to TIMTS institute who provided me with this great opportunity

CONTENTS

Purpose

To apply for a master's degree is a very expensive and intensive work. With this kernel, students will guess their capacities and they will decide whether to apply for a master's degree or not.

So, basically this set is about the Graduate Admissions data i.e. Given a set of standardized scores like GRE, TOEFL, SOP standard scores, LOR standard scores, what is probability (basically i have done a YES/NO scenario) of gaining admission into a particular school. All those folks who are preparing for MS, might point out this question, from where did you get SOP & LOR scores. These aren’t public figures ? I mean yes, it might not be public, but don’t you think universities might be grading these applications on some scale of rating so that the scores can be standardized. Hence the SOP, LOR scores.

Dataset

This dataset is created for prediction of graduate admissions and the dataset link is below:

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

Features in the dataset:

- GRE Scores (290 to 340)
- TOEFL Scores (92 to 120)
- University Rating (1 to 5)
- Statement of Purpose (1 to 5)
- Letter of Recommendation Strength (1 to 5)
- Undergraduate CGPA (6.8 to 9.92)
- Research Experience (0 or 1)
- Chance of Admit (0.34 to 0.97)

THEORY OF LINEAR REGRESSION

In machine learning, any problem can be taken as classification problem or regression problem. Different from classification, the prediction value is always continuous in a certain range, simply to say, target value is the probability of one event happening. For example, a medicine institution wants to diagnose a patient according to known a set of data of patients health records. If we want to know the patient is ill or not, this is a classification problem. But sometimes the result is not absolute. Before getting ill, the patient wants to know the probability of being ill, which can be considered as regression problem.

Linear regression is the simplest statistic model to imply the relation between variables. Firstly, let us only consider two variables relation, then I will extend it to multi variables linear regression .

A. Simple Linear Regression:-

With simple linear regression when we have a single input, we can use statistics to estimate the coefficients.

This requires that you calculate statistical properties from the data such as means, standard deviations, correlations and covariance. All of the data must be available to traverse and calculate statistics.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

$$y = B0 + B1*x$$

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

B. Multiple Linear Regression

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables. In essence,

multiple regression is the extension of ordinary least-squares (OLS) regression because it involves more than one explanatory variable.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

y_i=dependent variable

x_i=explanatory variables

β_0=y-intercept (constant term)

β_p=slope coefficients for each explanatory variable

ϵ=the model's error term (also known as the residuals)

IMPLEMENTATION

This work used Python programming for this project, as it is a high-level programming language and it has vast libraries and Python automates tasks and makes it efficient. Firstly, we need to install Python then we need to import some libraries, they are:

1) Numpy: Numpy is used for multi-dimensional arrays, it does element to element operations and it also has different methods for processing arrays.
2) Panda: Pandas is one of the highly used python libraries, it provides high performance. It manipulates data and it makes data analysis fast and easy.

3) Sklearn: It is most useful library, this library contains lot of efficient tools, it is used to build models like statistical modelling including classification, regression, clustering. After loading required packages, we divide dataset as training and testing as follows, here 80 % of dataset is taken as training and remaining 20 % as to perform test.

IMPORT LIBRARIES and LOAD DATA

First, let's import all the modules, functions and objects we are going to use.We can load the data directly from the UCI Machine Learning repository. We are using pandas to load the data. We will also use pandas next to explore the data both with descriptive statistics and data visualization.

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

```

```
# Data reading
```

In [2]:
```
raw_data=pd.read_csv("Admission_Predict.csv")
raw_data
```

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
...	...	...	...	...	...	...	...	...	...
395	396	324	110	3	3.5	3.5	9.04	1	0.82
396	397	325	107	3	3.0	3.5	9.11	1	0.84
397	398	330	116	4	5.0	4.5	9.45	1	0.91
398	399	312	103	3	3.5	4.0	8.78	0	0.67

Analysis the Data

In [3]:
```
1 raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial No.         400 non-null    int64  
 1   GRE Score          400 non-null    int64  
 2   TOEFL Score        400 non-null    int64  
 3   University Rating  400 non-null    int64  
 4   SOP                400 non-null    float64
 5   LOR                400 non-null    float64
 6   CGPA               400 non-null    float64
 7   Research           400 non-null    int64  
 8   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
```

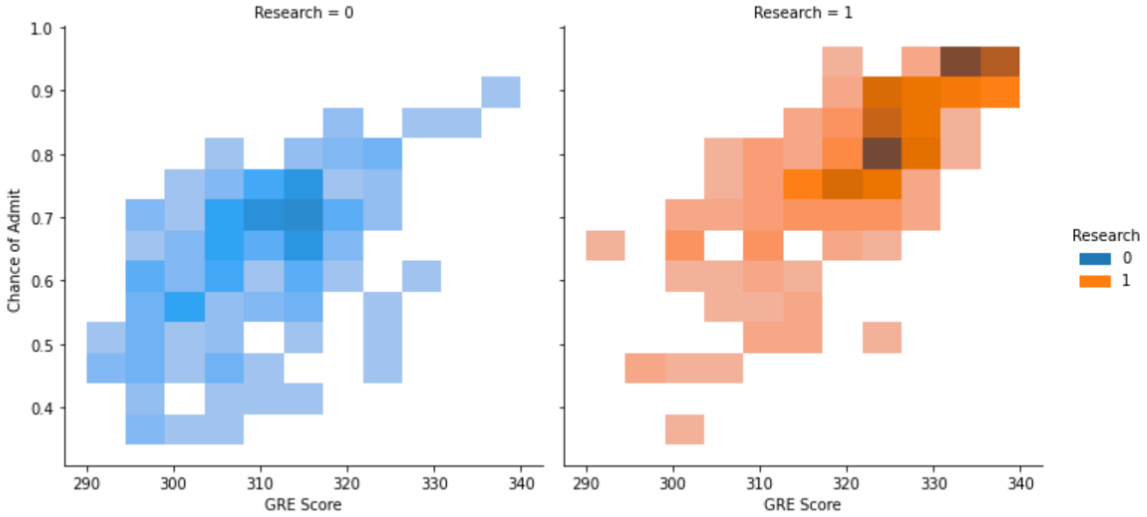
In [4]:
```
1 raw_data.describe()
```

Out[4]:

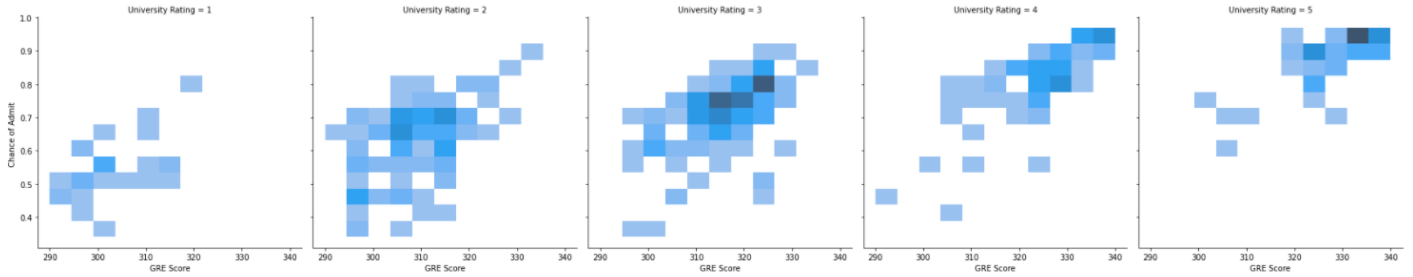
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	200.500000	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	115.614301	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

DATA VISUALIZATION

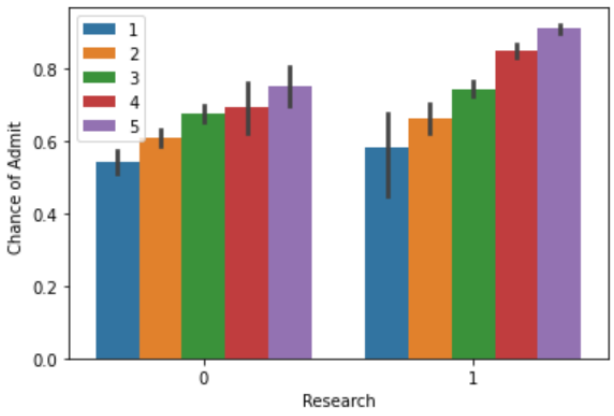
In [10]:
```
sns.displot(x="GRE Score",y="Chance of Admit ",data=df,hue="Research",col="Research")
plt.show()
```



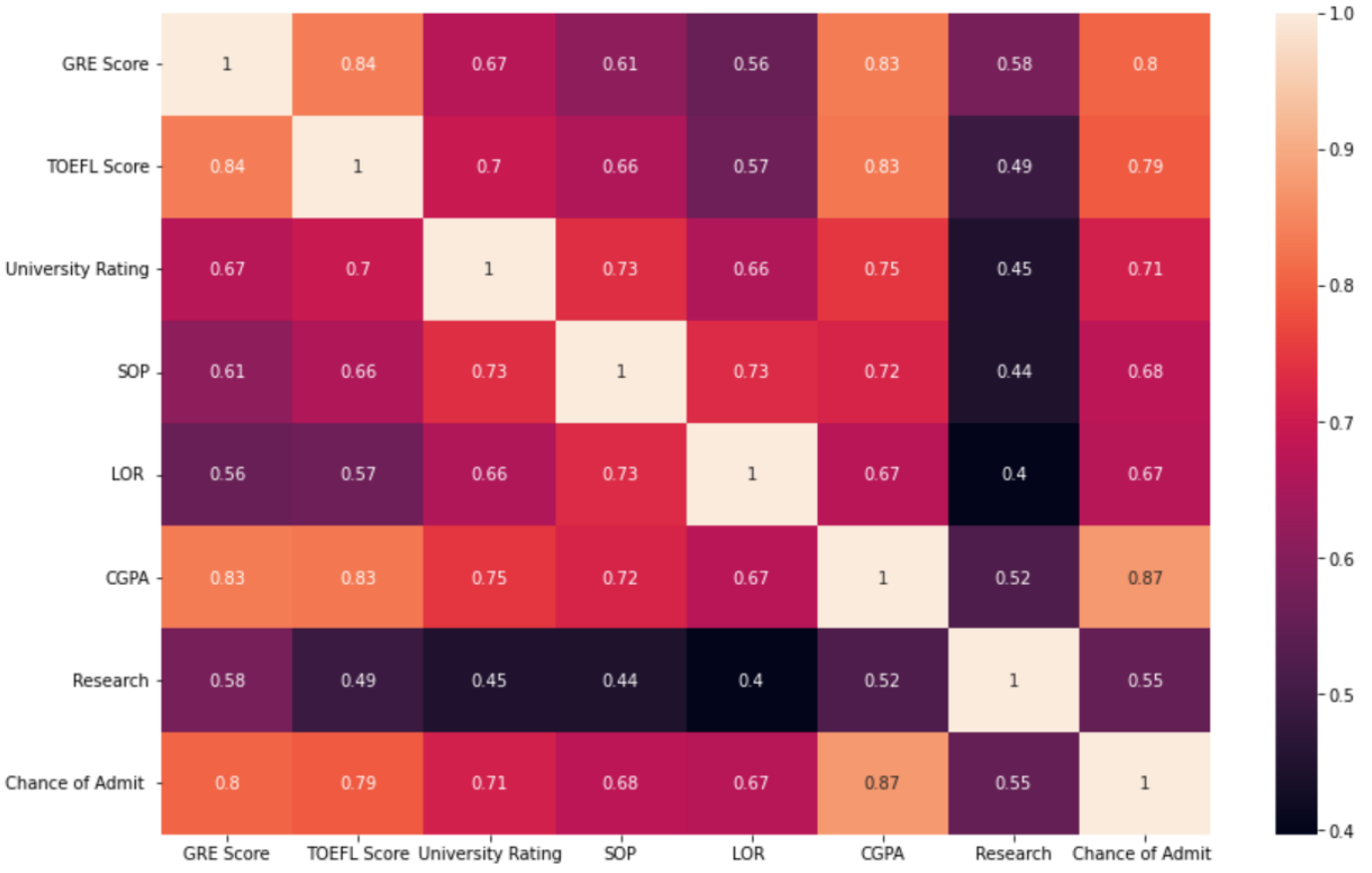
In [11]:
```
sns.displot(x="GRE Score",y="Chance of Admit ",data=df,col="University Rating")
plt.show()
```



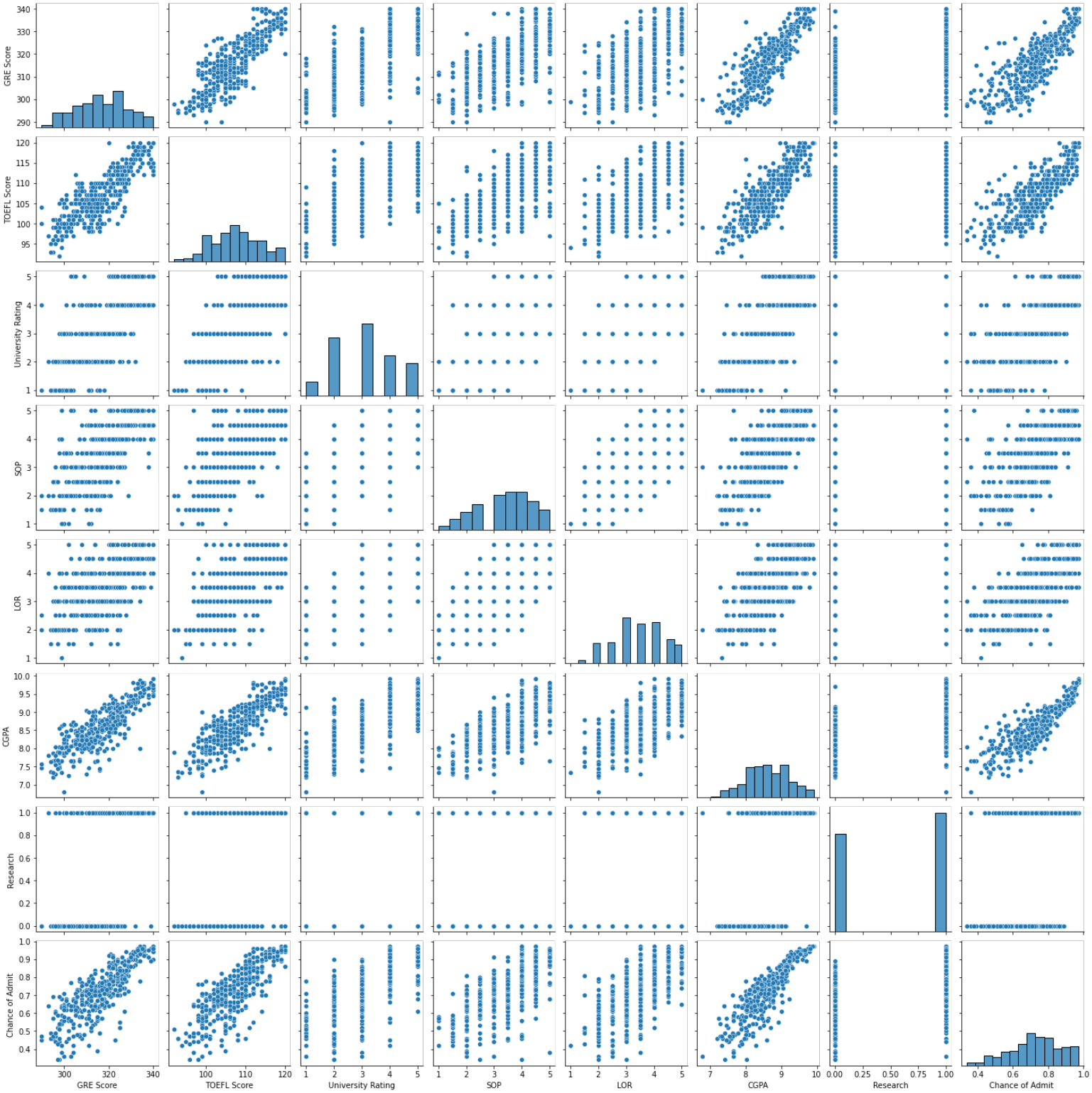
In [14]:
```
sns.barplot(x="Research",y="Chance of Admit ",data=df,hue="University Rating")
plt.legend(loc="upper left")
plt.show()
```



Heat Map:


GRE Score
TOEFL Score
University Rating
SOP
LOR
CGPA
Research
Chance of Admit
1 0.84 0.67 0.61 0.56 0.83 0.58 0.8
0.84 1 0.7 0.66 0.57 0.83 0.49 0.79
0.67 0.7 1 0.73 0.66 0.75 0.45 0.71
0.61 0.66 0.73 1 0.73 0.72 0.44 0.68
0.56 0.57 0.66 0.73 1 0.67 0.4 0.67
0.83 0.83 0.75 0.72 0.67 1 0.52 0.87
0.58 0.49 0.45 0.44 0.4 0.52 1 0.55
0.8 0.79 0.71 0.68 0.67 0.87 0.55 1
GRE Score
TOEFL Score
University Rating
SOP
LOR
CGPA
Research
Chance of Admit
1.0
0.9
0.8
0.7
0.6
0.5
0.4

Pari Plot


GRE Score
TOEFL Score
University Rating
SOP
LOR
CGPA
Research
Chance of Admit

TRAINING AND TESTING DATA

TRAINING AND TESTING DATA

In [16]:
```
X=df.drop('Chance of Admit ', axis=1)
y=df["Chance of Admit "]
```

In [17]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2, random_state=30)
```

In [19]:
```
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(320, 7)
(320,)
(80, 7)
(80,)
```

In [20]:
```
X_train.mean()
```

Out[20]:
```
GRE Score            316.618750
TOEFL Score          107.262500
University Rating      3.050000
SOP                    3.367188
LOR                    3.432812
CGPA                   8.591781
```

MODEL BUILDING

LINEAR REGRESSION

In [22]:
```
from sklearn.linear_model import LinearRegression
```

In [23]:
```
lr=LinearRegression()
```

In [24]:
```
lr.fit(X_train,y_train)
```

Out[24]:
```
LinearRegression()
```

In [27]:
```
lr.predict(X_test)
```

Out[27]:
```
array([0.86493903, 0.53541379, 0.82882584, 0.77449869, 0.70557862,
       0.93560153, 0.8565008 , 0.90485298, 0.7243016 , 0.70324605,
       0.94915833, 0.49841654, 0.83318167, 0.88587806, 0.95509146,
       0.73746878, 0.70702419, 0.92370053, 0.64908579, 0.53063033,
       0.64130877, 0.67835265, 0.60676069, 0.52637735, 0.6545361 ,
       0.51060514, 0.74580969, 0.52818007, 0.8469073 , 0.67911093,
       0.94971066, 0.6983349 , 0.60634218, 0.85858122, 0.82236274,
       0.62456135, 0.66937204, 0.74949407, 0.91249405, 0.60950727,
       0.7624889 , 0.87992015, 0.78021308, 0.58414467, 0.73768712,
       0.94051552, 0.74086916, 0.7440317 , 0.78138668, 0.92725736,
       0.57434778, 0.73185692, 0.6376929 , 0.76914511, 0.61295748,
       0.53486453, 0.72677789, 0.92839598, 0.8275333 , 0.62995308,
       0.78586624, 0.64332399, 0.69408698, 0.79672477, 0.77594233,
       0.81635791, 0.74914181, 0.84424656, 0.67132881, 0.77333738,
       0.78420481, 0.65769607, 0.56215354, 0.90142445, 0.68405328,
       0.87690236, 0.7925973 , 0.49893607, 0.47352703, 0.83491779])
```

In [28]:
```
y_test
```

Out[28]:
```
35     0.88
316    0.54
281    0.80
74     0.74
296    0.76
       ...
188    0.93
245    0.81
118    0.47
272    0.49
365    0.86
Name: Chance of Admit , Length: 80, dtype: float64
```

In [30]:
```
lr.score(X_test,y_test)*100
```

Out[30]:
```
83.25349083642278
```

CONCLUSION

We got a accuracy of about 83 % using Linear Regression.

CONCLUSION

My project mainly includes three parts: data preprocessing, base model selection, modeling and stacking. Because the original data is completed and clean without too many features, most energy I spent on my project is the base model selection and stacking. Selecting base model is important so as to I save more energy in result analysis and have base line to compare with following models.

The Analysis done on the data helps both student and college to choose according to the grades.

Without proper analysis student will not be able to choose college, which affect the decision making to which they apply. That’s where Data Analysis help them to select college according to their marks.

REFERENCES

- https://machinelearningmastery.com/linear-regression-for-machine-learning
- https://www.kaggle.com/mohansacharya/graduate-admissions