

# **UNIVERSITY ADMIT ELIGIBILITY PREDICTOR**

## **PROJECT REPORT**

**TEAM ID: PNT2022TMID05170**

**Submitted by:**

**DEVADHARSHINI M \_921319106043**

**DHARANI V \_921319106048**

**DIVYA J \_921319106051**

**DIVYA BHARATHI S \_921319106052**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**AT**

**PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

**DINDIGUL- 624 622**

# CONTENTS

## **1. Introduction**

- 1.1 Project Overview
- 1.2 Purpose

## **2. Literature Survey**

- 2.1 Existing Problem
- 2.2 References
- 2.3 Problem Statement Definitions

## **3. Ideation & Proposed Solution**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution Fit

## **4. Requirement Analysis**

- 4.1 Functional Requirement
- 4.2 Non-Functional Requirement

## **5. Project Design**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

## **6. Project Planning & Scheduling**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule

## **7. Coding & Solutioning**

- 7.1 Feature 1
- 7.2 Feature 2

## **8. Testing**

- 8.1 Test Cases
- 8.2 User Acceptance Testing

## **9. Results**

- 9.1 Performance Metrics

## **10. Advantages & Disadvantages**

## **11. Conclusion**

## **12. Future Scope**

## **13. Appendix**

13.1 Source Code

13.2 GitHub

13.3 Project Demo Link

# 1. INTRODUCTION

The world's business sector is escalating and is constantly seeking information and experiences that are commonly beneficial to individuals. Young specialists who need to stay in their current positions are always looking for advanced degrees to help them address their skills and information. As such, the number of her sophomores applying for graduation exams has increased over the past decade. One of her main concerns is getting into fantasy her university. You can see that undergraduates are actually choosing to get their education at prestigious universities. Furthermore, when it comes to international alumni, the United States is the main trend for most of them. The most prestigious universities offer a wide range of courses accessible in any order, exceptionally accredited teaching and education programs, an international second Research scholarships for degrees are available.

According to Gauges, more than 4,444 of her 10 million international sophomores are enrolled in her 4,200+ colleges and universities, both private and public. In general, the number of undergraduates concentrated in America comes from Asian countries such as India, Pakistan, Sri Lanka, Japan and China. Select the United Kingdom, Germany, Italy, Australia, Canada as well as the United States. These countries are witnessing a rapid increase in the number of individuals seeking more advanced investigations. The basic reason why sophomores go on to master's programs in foreign graduate schools is that the number of vacancies is low and the number of people in these positions in each country is huge. This has led many professional undergraduates to pursue postgraduate studies. You can see that there are quite a few bachelor's degrees and master's degrees in computer science at US universities. The focus of this study applies to these undergraduate degrees. Many schools in the US follow comparative requirements for

undergraduate accreditation. Schools consider several variables, including placement in fitness assessments and school performance ratings. English rankings are determined by exposure in English proficiency tests such as TOEFL and IELTS.

The University's Admissions Advisory Board makes decisions regarding the acceptance or rejection of specific young researchers based on the general profile of the applicant's application. Records recorded with this company are marked with informative areas. Acknowledgment is a 400-row data set containing seven different autonomic factors. ie

- Graduate Record Examination 1 (GRE) score. The score consists of 340 foci.
- English as a Foreign Language (TOEFL) test score. It consists of 120 priority areas.
- Uni.Rating. Shows the position of colleges offering bachelor's degrees among various colleges. Your score will be out of 5.
- Statement of Purpose (SOP), a record written to reveal the life, motivations and inspirations of a selected degree/college applicant. The score consists of five focal points.
- The strength of a letter of recommendation (LOR) verifies the applicant's professional experience, falsifies validity, supports certainty, and guarantees your competence. The score consists of five focal points.
- Undergraduate GPA (CGPA) from 10.
- Research experience (either 0 or 1) that could support the application, such as distributing research papers at conferences or filling out as a right-hand exam for university faculty. One ward variable can be anticipated which is possibility of affirmation, that is as per the input given will be going from 0 to 1.

## 1.1 PROJECT OVERVIEW

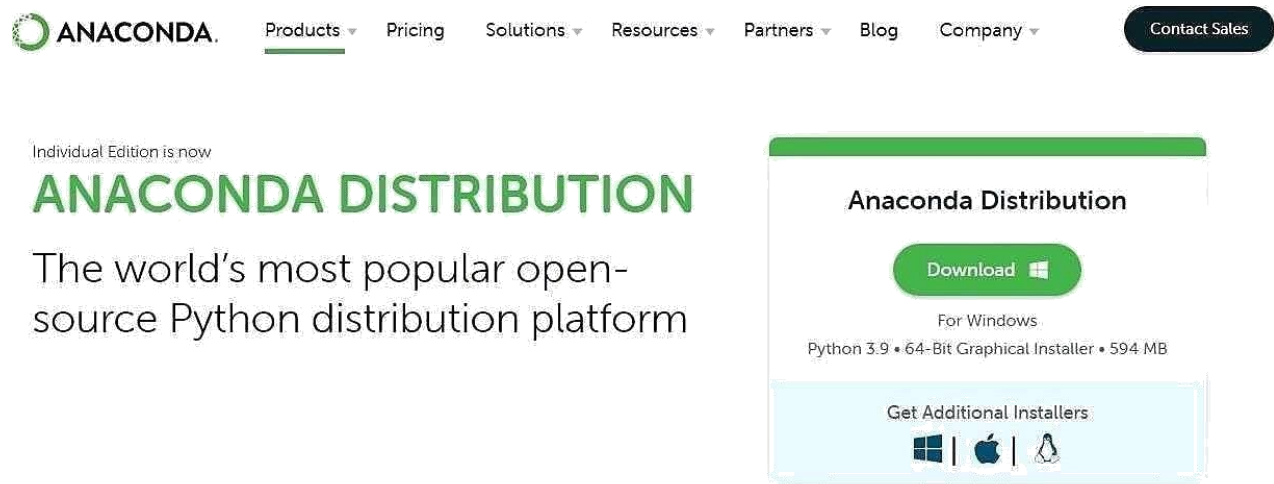
### PRE-REQUISITIES

#### Anaconda Installation:

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data science packages suitable for Windows, Linux, and macOS. Developed and maintained by Anaconda. Founded in 2012 by Peter Wang and Travis Olyphant. As Anaconda, also known as Anaconda Distribution or Anaconda Individual Edition, the company's other products include his Anaconda Team Edition and Anaconda Enterprise Edition, neither of which are free.

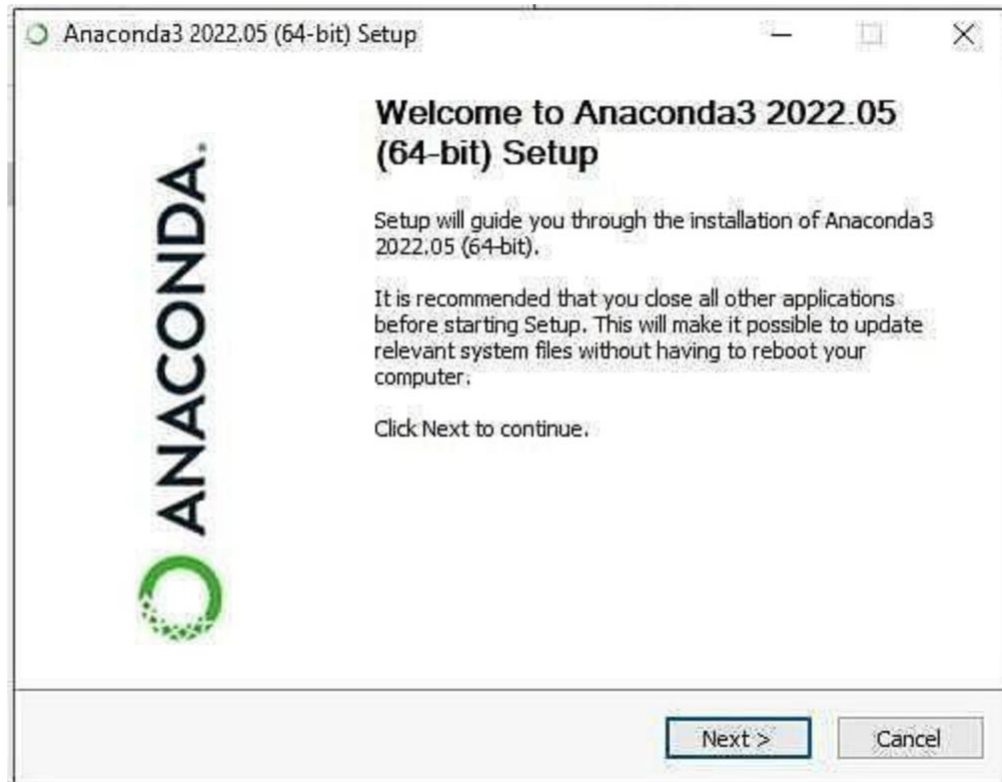
#### WAY TO INSTALL ANACONDA:

##### STEP 1: Download and Anaconda

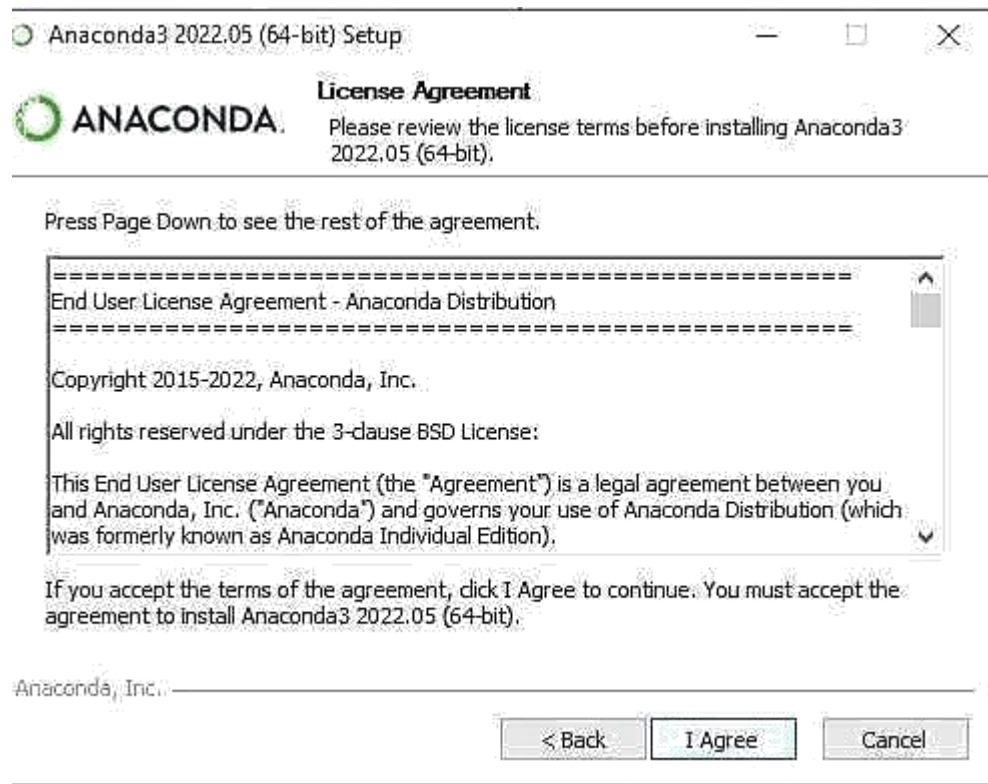


The screenshot shows the Anaconda website. At the top is the Anaconda logo and a navigation bar with links: Products, Pricing, Solutions, Resources, Partners, Blog, and Company. A 'Contact Sales' button is on the right. The main content area features the text 'Individual Edition is now' followed by 'ANACONDA DISTRIBUTION' in large green letters. Below this is the tagline 'The world's most popular open-source Python distribution platform'. On the right, there is a 'Download' button with a Windows icon, labeled 'For Windows' and 'Python 3.9 • 64-Bit Graphical Installer • 594 MB'. At the bottom, there is a section 'Get Additional Installers' with icons for Windows, Apple, and Linux.

##### STEP 2: Install the Anaconda

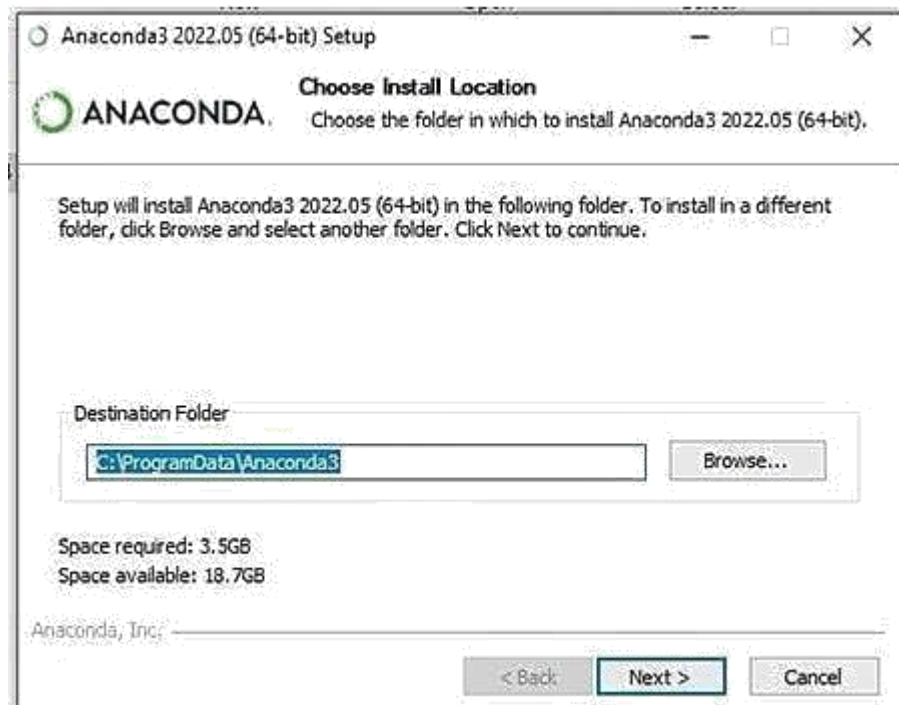


### STEP 3: Click I Agree

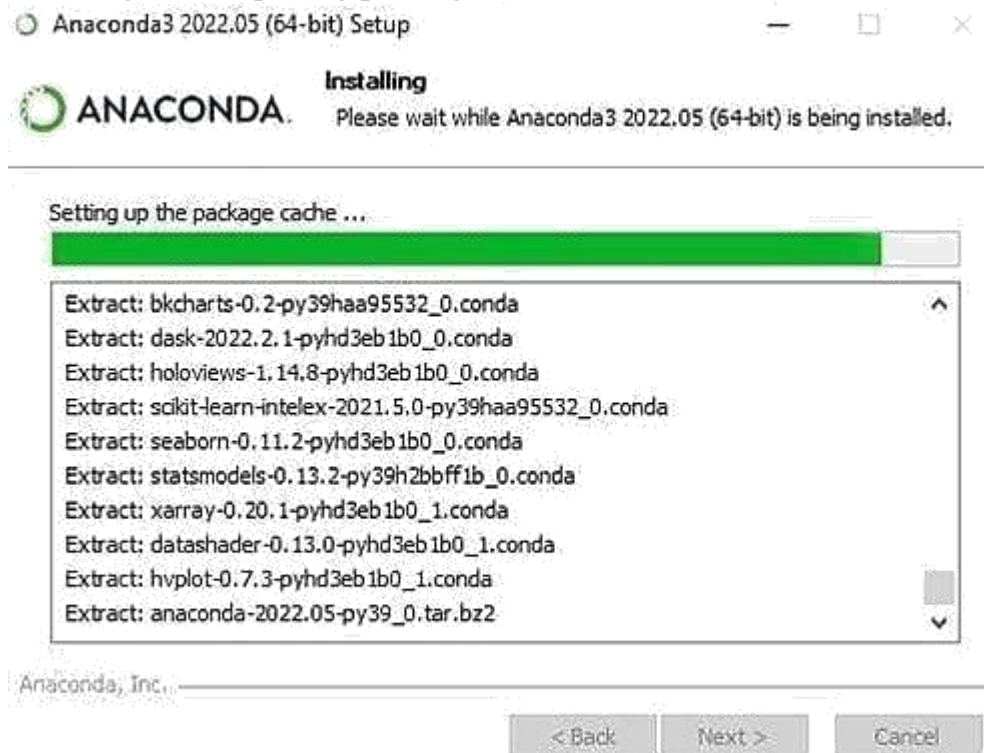




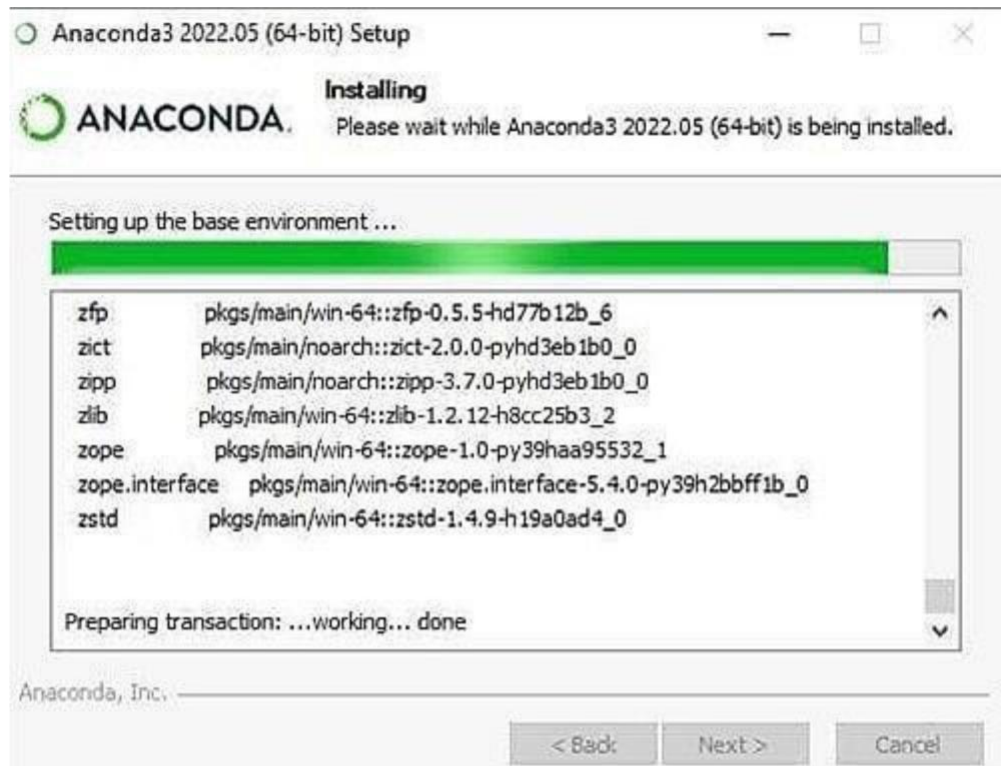
## STEP 4: Choose the Installation Location



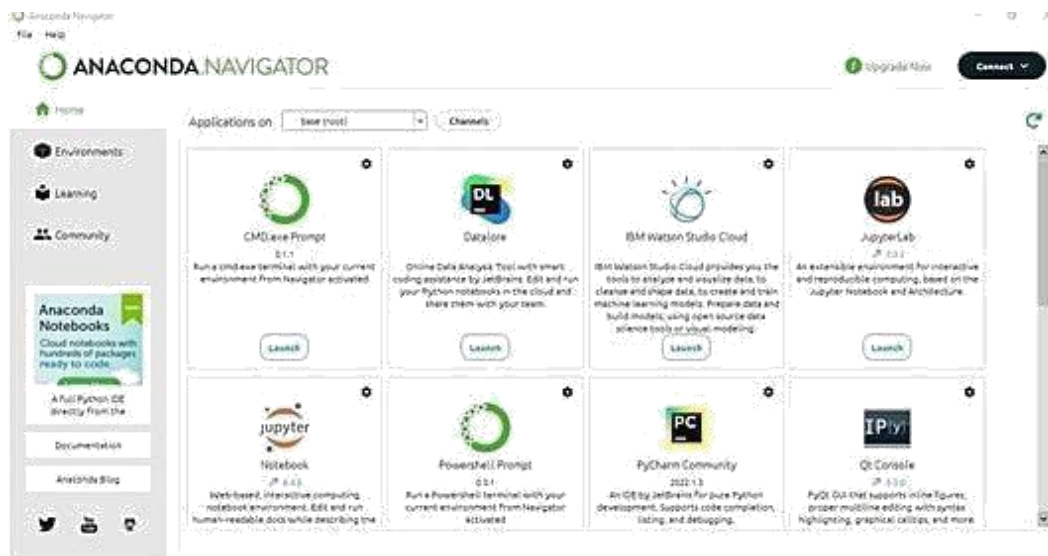
## STEP 5: Installing the Requiring packages



## STEP 6: Setting up the base environment

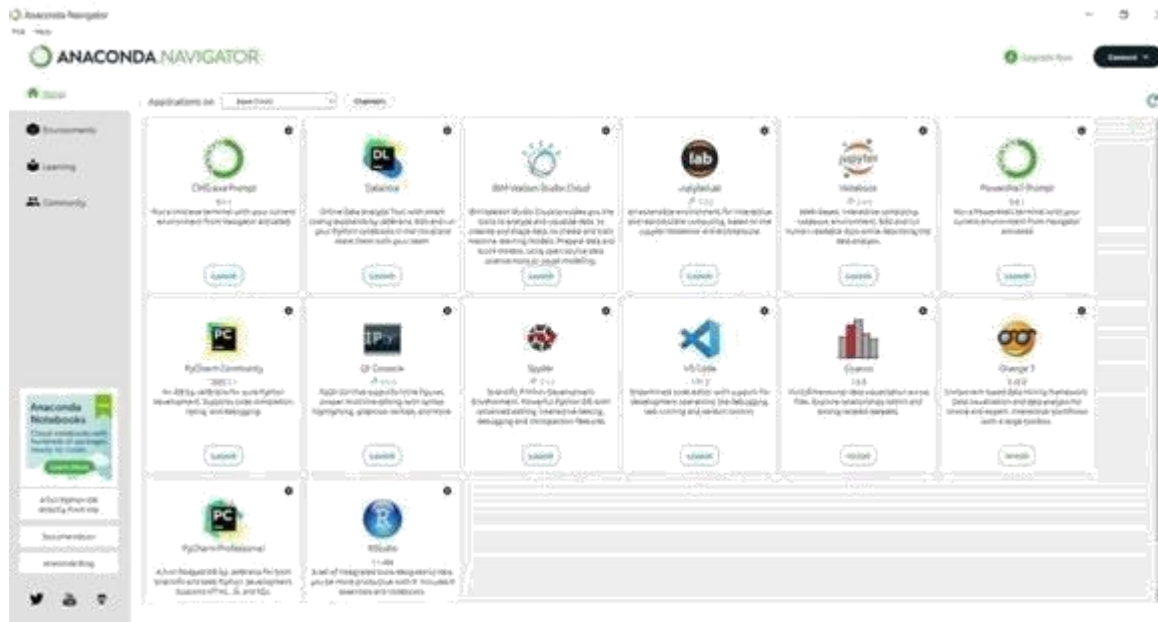


**STEP 7:** Successfully Installed and check the Anaconda Navigator working or not

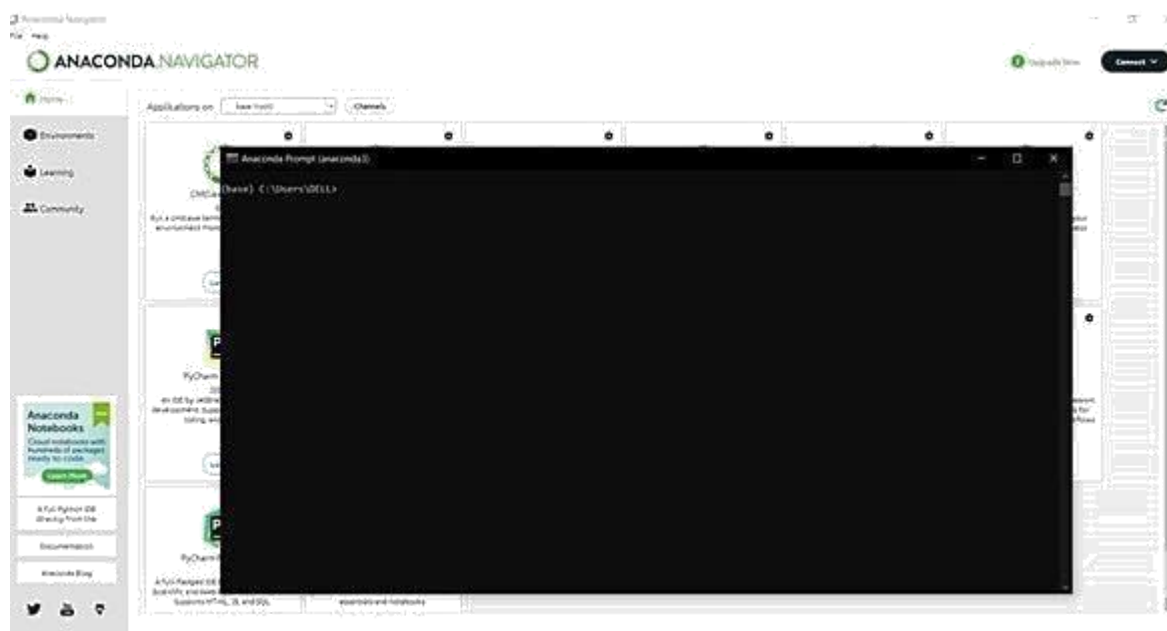


## Python packages installation:

**Step 1:** Open the anaconda navigator in the start menu



**Step 2:** Open the CMD.exe prompt



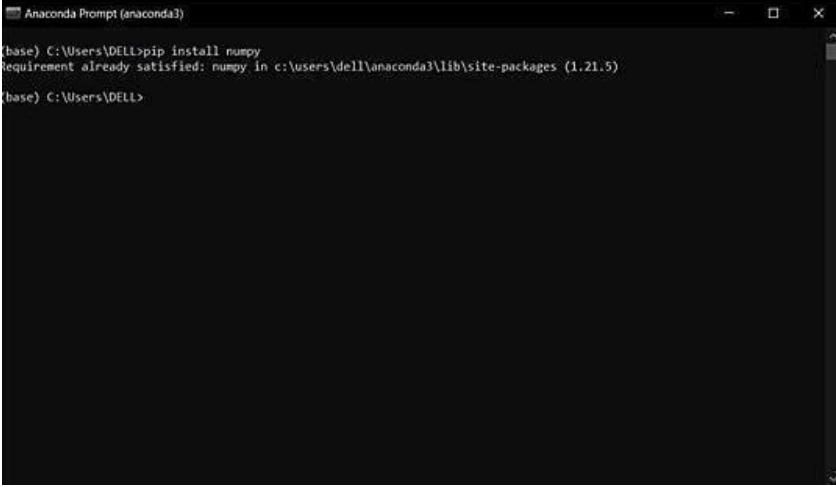
### Step 3: Install the NUMPY package

To enter the numpy package enter the command in the CMD.exe

Command: **Pip install numpy**

#### **Numpy:**

This package is used to perform numerical computations. This package comes pre-installed with Anaconda. NumPy is used for manipulating arrays. NumPy stands for Numerical Python.



```
Anaconda Prompt (anaconda3)
(base) C:\Users\DELL>pip install numpy
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (1.21.5)
(base) C:\Users\DELL>
```

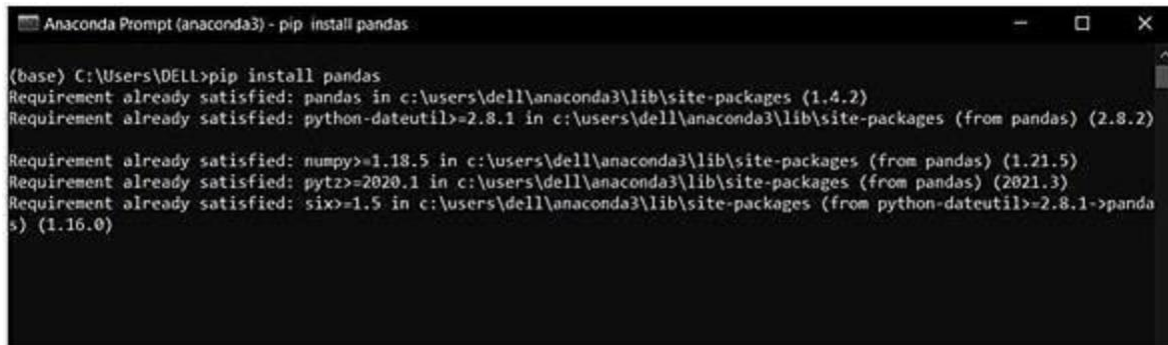
### Step 4: Install the pandas package.

To enter the pandas package enter the command in the CMD.exe

Command: **Pip install pandas**

#### **Pandas:**

Pandas is one of the most widely used Python libraries for data science. It provides powerful and easy-to-use structure and data analysis tools. This package comes pre-installed with Anaconda. An open source library built on top of the NumPy library. A Python package that provides various data structures and operations for working with numerical data and time series. Mainly, it's common for data to be imported and analyzed much easier. Pandas is fast, providing users with high performance and productivity.



```
Anaconda Prompt (anaconda3) - pip install pandas

(base) C:\Users\DELL>pip install pandas
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

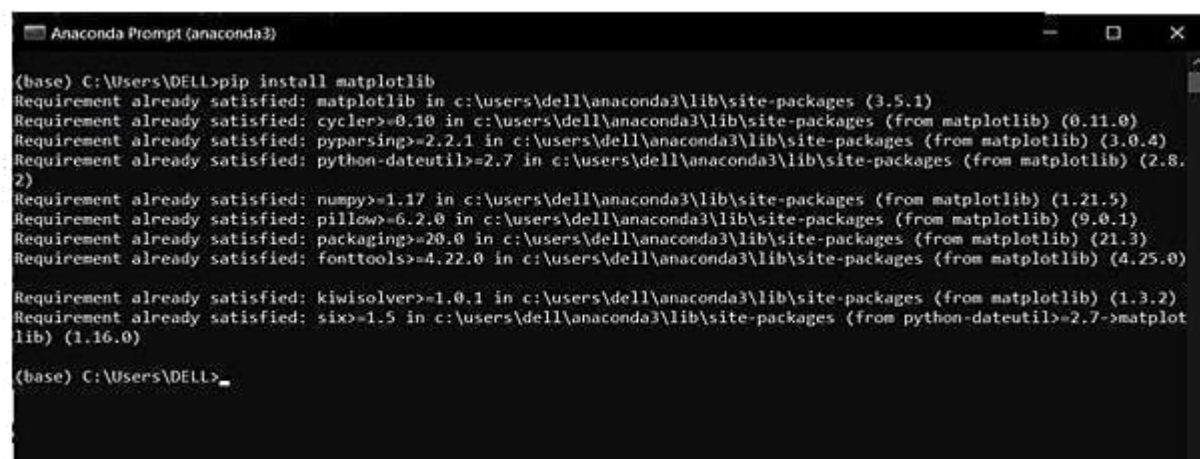
## Step 5: Install the Matplotlib package.

To enter the Matplotlib package enter the command in the CMD.exe

Command: **Pip install Matplotlib**

### Matplotlib:

Matplotlib is a comprehensive library for creating static, animated and interactive visualizations in Python. This package comes pre-installed with Anaconda. Matplotlib is a nice visualization library in Python for 2D plotting of arrays. Matplotlib is a cross-platform data visualization library based on NumPy arrays and designed to work with the wider SciPy stack. Introduced by John Hunter in 2002.



```
Anaconda Prompt (anaconda3)

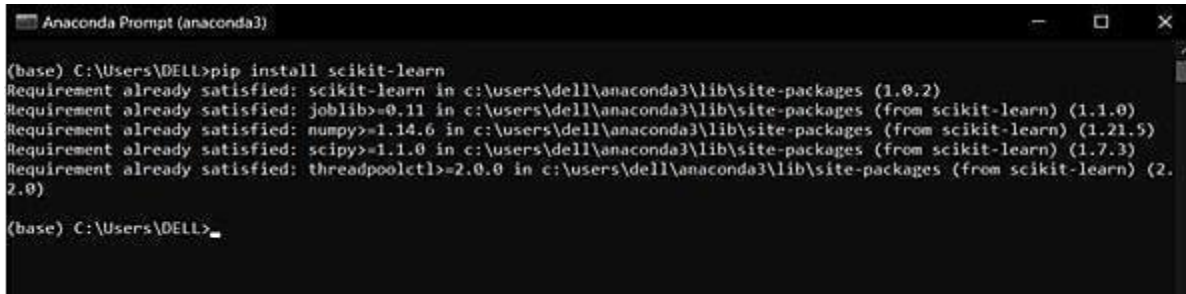
(base) C:\Users\DELL>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

(base) C:\Users\DELL>
```

## Step 6: Install the Scikit-learn package.

To enter the Scikit-learn package enter the command in the CMD.exe

This is a machine learning library for the Python programming language. This package comes pre-installed with Anaconda. Scikit Learn in Python is primarily used to focus on modeling in Python. It was only focused on modeling, not loading data.

A screenshot of the Anaconda Prompt window. The title bar reads 'Anaconda Prompt (anaconda3)'. The command prompt shows the command '(base) C:\Users\DELL>pip install scikit-learn'. The output lists several requirements already satisfied: 'scikit-learn' (1.0.2), 'joblib' (0.11), 'numpy' (1.14.6), 'scipy' (1.1.0), and 'threadpoolctl' (2.0.0). The prompt ends with '(base) C:\Users\DELL>'.

```
Anaconda Prompt (anaconda3)

(base) C:\Users\DELL>pip install scikit-learn
Requirement already satisfied: scikit-learn in c:\users\dell\anaconda3\lib\site-packages (1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.21.5)
Requirement already satisfied: scipy>=1.1.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.7.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)

(base) C:\Users\DELL>
```

**Step 7:** Install the Flask package.

To enter the Flask package enter the command in the CMD.exe

Command: **Pip install Flask**

**Flask:**

Flask is a lightweight WSGI web application framework Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts called Pocco. Flask is based on the WSGI toolkit tools and the Jinja2 template engine. Both are Pocco projects.

A screenshot of the Anaconda Prompt window. The title bar reads 'Anaconda Prompt (anaconda3)'. The command prompt shows the command '(base) C:\Users\DELL>pip install flask'. The output lists several requirements already satisfied: 'flask' (1.1.2), 'click' (5.1), 'werkzeug' (0.15), 'jinja2' (2.10.1), 'itsdangerous' (0.24), 'colorama' (0.4.4), and 'MarkupSafe' (0.23). The prompt ends with '(base) C:\Users\DELL>'.

```
Anaconda Prompt (anaconda3)

(base) C:\Users\DELL>pip install flask
Requirement already satisfied: flask in c:\users\dell\anaconda3\lib\site-packages (1.1.2)
Requirement already satisfied: click>=5.1 in c:\users\dell\anaconda3\lib\site-packages (from flask) (8.0.4)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.0.3)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.11.3)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: colorama in c:\users\dell\anaconda3\lib\site-packages (from click>=5.1->flask) (0.4.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\dell\anaconda3\lib\site-packages (from Jinja2>=2.10.1->flask) (2.0.1)

(base) C:\Users\DELL>
```

## PROJECT FLOW

You will go through all the steps mentioned below to complete the project.

- User interacts with the UI (User Interface) to enter Data
- The entered data is analyzed by the model which is integrated
- Once model analyses the input the prediction is showcased

on the UI To accomplish this, we have to complete all the activities and tasks listed below

- Data Collection.
  - Collect the dataset or Create the dataset
- Data Preprocessing.
  - Import the Libraries.
  - Importing the dataset.
  - Checking for Null Values.
  - Data Visualization.
  - Taking care of Missing Data.
  - Label encoding.
  - One Hot Encoding.
  - Feature Scaling.
  - Splitting Data into Train and Test.
- Model Building
  - Training and testing the model
  - Evaluation of Model
- Application Building
  - Create an HTML file
  - Build a Python Code

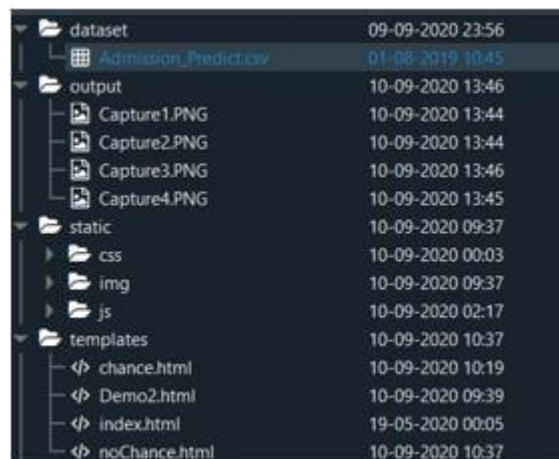


## PROJECT OBJECTIVES

- To understand regression and classification problems.
- To grab insights from data through visualization.
- Applying different Machine Learning algorithms to determine the probability of acceptance in a particular university.
- Evaluation metrics build a web application using the Flask framework

## PROJECT STRUCTURE

Create a Project folder that contains files as shown below



## DATA COLLECTION

The path to common information varies by project type. ML projects use real-time information. Information indexes can be collected from a variety of sources such as documents, data sets, sensors, and other sources, using free information collection from the Internet. Kaggle and the UCI Machinelearning Repository are the most commonly used repositories for sorting.



Collect the dataset or Create the dataset.

481 lines (481 sloc) | 12.6 KB

Raw Blame    

Q Search this file...

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1									
2	1	337	118	4	4.5	4.5	9.65	1	0.92
3	2	324	107	4	4	4.5	8.87	1	0.76
4	3	316	104	3	3	3.5	8	1	0.72
5	4	322	110	3	3.5	2.5	8.67	1	0.8
6	5	314	103	2	2	3	8.21	0	0.65
7	6	330	115	5	4.5	3	9.34	1	0.9
8	7	321	109	3	3	4	8.2	1	0.75
9	8	308	101	2	3	4	7.9	0	0.68
10	9	302	102	1	2	1.5	8	0	0.5
11	10	323	108	3	3.5	3	8.6	0	0.45
12	11	325	106	3	3.5	4	8.4	1	0.52
13	12	327	111	4	4	4.5	9	1	0.84
14	13	328	112	4	4	4.5	9.1	1	0.78
15	14	307	109	3	4	3	8	1	0.62
16	15	311	104	3	3.5	2	8.2	1	0.61
17	16	314	105	3	3.5	2.5	8.3	0	0.54
18	17	317	107	3	4	3	8.7	0	0.66
19	18	319	106	3	4	3	8	1	0.65
20	19	318	110	3	4	3	8.8	0	0.63
21	20	303	102	3	3.5	3	8.5	0	0.62
22	21	312	107	3	3	2	7.9	1	0.64
23	22	325	114	4	3	2	8.4	0	0.7
24	23	328	116	5	5	5	9.5	1	0.94
25	24	334	119	5	5	4.5	9.7	1	0.95
26	25	336	119	5	4	3.5	9.8	1	0.97

## DATA PRE-PROCESSING

### Importing the Libraries:

- It is important to import all the necessary libraries such as pandas, numpy, matplotlib.

- **Numpy** : It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
- **Pandas**: It is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.
- **Seaborn**: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **Matplotlib**: Visualisation with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python

## Reading the Datasets:

1. You might have your data in .csv files, .excel files

Let's load a .csv data file into pandas using `read_csv()` function. We will need to locate the directory of the CSV file at first

```
#read_csv is a pandas function to read csv files
data = pd.read_csv('Admission_Predict.csv')
```

If your dataset is in some other location ,Then see below

command Data=

```
pd.read_csv(r"File_location/filename.csv")
```

Note: r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

Our Dataset Admission\_Predict contains following  
Columns 1. Serial No.

2. GRE Score
3. TOEFL Score
4. University Rating
5. SOP
6. LOR
7. CGPA
8. Chance of Admit

### Handling Missing Values:

After loading it is important to check the complete information of data as it can indicate many of the hidden information such as null values in a column or a row

Check for the null values. if it is present then the following steps can be performed

- Imputing data using the Imputation method in sklearn.
- Filling NaNvalues with mean, median, and mode using fillna() method. You can check the null values with the function isnull().any()

```
data.isnull().any()
```

```
GRE Score      False
TOEFL Score    False
University Rating False
SOP            False
LOR            False
CGPA           False
Research       False
Chance of Admit False
dtype: bool
```

If the dataset contains null values then the above functions return as true. But if you look at the dataset you can observe that the dataset does not have any null values.

You can also check the number of null values present in the columns by the using `isnull().sum()` function. As we don't have categorical data then we can skip the steps of label encoding and one-hot encoding

### **Data Visualization:**

Data visualization is where a given dataset is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data. Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. Machine learning models will perform poorly on data that wasn't visualized and understood properly.

To visualize the dataset we need libraries called Matplotlib and Seaborn. The Matplotlib library is a Python 2D plotting library that allows you to generate plots, scatter plots, histograms, bar charts etc.

### **Splitting Dependent And Independent Columns:**

We need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

- To read the columns, we will use `iloc` of pandas (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

Let's split our dataset into independent and dependent variables.

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

```
y=data.iloc[:,7:].values  
y
```

From the above code “:” indicates that you are considering all the rows in the dataset and “0:7” indicates that you are considering columns 0 to 7 such as year, month, and day as input values and assigning them to variable x. In the same way in the second line “:” indicates you are considering all the rows and “7:” indicates that you are considering only the last column as output value and assigning them to variable y.

Let's Check the shape of x and Y

```
x.shape
```

```
(1991, 7)
```

```
y.shape
```

```
(1991, 1)
```

- You can see in x we have 1991 rows with 7 columns and y has 1 column with the same number of rows

**Splitting The Data Into Train And Test:**

To train the model, first split the model into two segments: "training data" and "testing data". The classifier is trained using a 'training data set' and the performance of the classifier is tested on a non-fitting 'test data set'.

**Training Set:** The training Set is material for computers to learn how to process data. The AI uses computation to do the training part. The training dataset is used to learn and tune the classifier parameters.

**Test set:** A set of unseen data used solely to evaluate the performance of the fully displayed classifier.

When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test dataset. For this, you will need a dataset that is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

- Train Dataset: Used to fit the machine learning model.
- Test Dataset: Used to evaluate the fit machine learning model.

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size = 0.2,random_state = 10)
```

In general, you can allocate 80% of the dataset to the training set and the remaining 20% to the test set. We will create 4 sets

1. x\_train
2. x\_test
3. y\_train
4. y\_test

There are a few other parameters that we need to understand before we use the class:

- `test_size`: this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset and remaining a train dataset
- `random_state`: here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the `Random_state` class, which will become the number generator. If you don't pass anything, the `Random_state` instance used by `np.random` will be used instead.

## 1.2 PURPOSE

This is the project for a new web-based University Admit Eligibility Predictor. Predictor is an ML based application that asks for the users to input their academic transcripts data and calculates their chances of admission into the University Tier that they selected. It also provides an analysis of the data and shows how chances of admissions can depend on various factors. This document describes the scope, objectives and goals of the system. In addition to describing the non-functional requirements, this document models the functional requirements with use cases, interaction diagrams and class models. This document is intended to direct the design and implementation of the target system in an object-oriented language.

## **2. LITERATURE SURVEY 2.1**

### **Existing Problem**

It's almost admission season and I've couple of friends who are in panic mode waiting for a call from the universities they've applied at.

This made me think — How can we predict whether a student will get an admit or not? What are the parameters for selection? Can it be mathematically expressed?

All of these questions started popping up. This is the main existing problem.

## **2.2 References**

### **A UNIVERSITY ADMISSION PREDICTION SYSTEM USING STACKED ENSEMBLE LEARNING**

**AUTHOR:** S. Sridhar, S. Mootha and S. Kolagati **OVERVIEW:** "A University Admission Prediction System using Stacked Ensemble Learning," 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), 2020, pp. 162-167, doi: 10.1109/ACCTHPA49271.2020.9213205. **Abstract:** For an aspiring graduate student, shortlisting the universities to apply is a difficult problem. Since an application is extremely dynamic, students often tend to wonder if their profile matches the requirement of a certain university. Moreover, the cost of applying to a university is extremely high making it critical that students shortlist universities based on their profile. A university admission prediction system is quite useful for students to determine their chances of acceptance to a specific university. The system could make use of data related to previous applicants to various universities and their admit or reject status. Earlier models of such prediction systems suffer from several drawbacks such as not considering important parameters like GRE (Graduate Record Exam) scores or research experience. Further, the accuracy reported by earlier models is also not sufficiently high. In this paper, a stacked ensemble model that predicts the chances of admit of a student to a particular university has been proposed. The proposed model takes into consideration various factors related to the student including their research experience, industry experience etc. Further, the system proposed has been evaluated against various other machine learning algorithms including other deep learning methods. It is observed that the proposed model easily outperforms all other models and provides a very high accuracy. **Observation:** An effective method has been proposed to predict the chances of a student being admitted to a specific university. In addition, we have compared the performance of various machine learning algorithms to



the proposed method in predicting admits.

ACCURACY: 91%

URL: <https://ieeexplore.ieee.org/document/9213205>

### **TITLE: PREDICTION PROBABILITY OF GETTING AN ADMISSION INTO A UNIVERSITY USING ML**

AUTHOR: A. Sivasangari, V. Shivani, Y. Bindhu, D. Deepa, R. Vignesh OVERVIEW:

In the present conditions, students regularly have difficulty finding a fitting institution to pursue higher studies based on their profile. There are some advisory administrations and online apps that recommend universities but they ask huge consultancy fees and online apps are not accurate. So, the aim of this research is to develop a model that predict the percentage of chances into the university accurately. This model provides also the analysis of scores versus chance of prediction based on historical data so that students can understand whether their profile is suitable or not. The proposed model uses linear regression and random forest algorithms but cat boost algorithm is giving highest accuracy. ACCURACY: 95%

URL: <https://ieeexplore.ieee.org/document/9418279>

### **TITLE: PREDICTING THE POST GRADUATE ADMISSIONS USING CLASSIFICATION TECHNIQUES**

AUTHOR: Selvaprabu Jeganathan, Saravanan Parthasarathy and P. M. Ashok Kumar

OVERVIEW: Decision making by applying data mining methods is being used in many service organizations. Educational bodies gradually started to use the business intelligence techniques to identify the current progress in their institutions. Numerous factors which have an impact in academia will be vivid to the educationalists while applying data mining techniques on the academic data. By employing the data mining methodologies, we could identify different patterns which aid institutions to take strategic decisions to improve the students' academic performance. Potential graduate students will have a dilemma on identifying the universities for their post graduate admissions and on the other hand an average graduate student would be uncertain on getting post graduate admission in a reputed university based on their academic scores. In this study, we applied the classification techniques such as Logistic Regression, KNN Classification, Support Vector Classification, Naive Bayes Classification, Decision Tree Classification and Random Forest Classification on the given academic admission dataset.

ACCURACY: 99%

URL: <https://ieeexplore.ieee.org/document/939681>

## 2.3 Problem Statement Definition

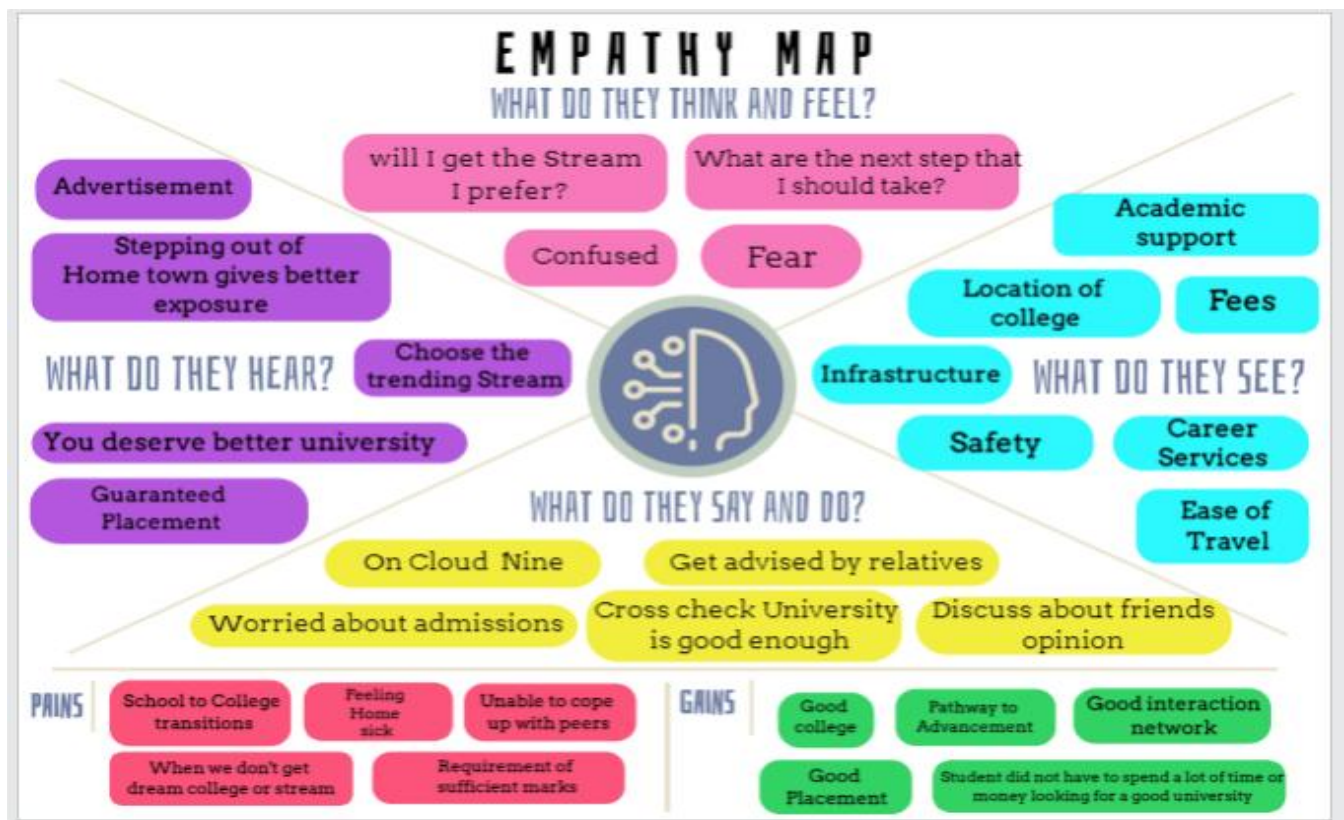
<b>Problem Statement</b>	<b>I am (Customer)</b>	<b>I'm trying to</b>	<b>But</b>	<b>Because</b>	<b>Which makes me feel</b>
PS-1	A student	Enroll for a PG course in the University	I am unaware of eligible cut off	The calculation for the cut off is not available in the website	Getting confused to admit university to get admission for different Universities
PS-2	Administator	To download all the data from the students	Details furnished by the students are incomplete	There is the problem in the seerver because of huge volume if data	Frustated and not able to download the documents

### 3. IDEATION AND PROPOSED SOLUTION

Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process. In this project the ideation phase consist of,

- Empathy Map
- Brainstorming
- Proposed Solution
- Problem Solution Fit

#### 3.1 Empathy Map



## 3.2 Ideation & Brainstorming

### Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

**TIP**  
You can select a sticky note and hit the pencil (which is next to) icon to start drawing!

#### Keerthana S

Students below 18yrs cannot apply for any streams.

Students applying for Polytechnic studies should be minimum of 16 yrs.

Students Profile should match with their originals.

If their profile details are not matched, their admission will be rejected.

#### Aarthikha R

If their marks are below the expectation of University, their application will be rejected.

If they are PG applicants, then their UG Certificates have to be submitted.

All the required certificates, proofs have to be submitted during registration.

Original certificates of students have to be maintained as Confidential.

#### Nisha M

Marks of entrance exams is considered.

Required Certificates have to be submitted if the applicants is a handicapped person or specially abled.

Students who are applying for Sports Quota have to submit their Sports certificates.

Student's Tenth, Twelfth marks, CGPA for PG applicants have to be collected.

#### Rabeka S

Certificate proof has to be uploaded if they are First Graduates.

Students under reservation quota are given more preference for admission.

Based on their scores, branches of University will vary.

Proof of entrance exams specific to their applied course has to be submitted.

#### Anuvarshini G

The cutoff requirement of admission is done based on caste.

Admissions are done based on the marks obtained by students in University level Entrance Exams.

According to their suggestions, specially abled people are given location.

Students from other states receive preference over those with specially abled.

### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

**TIP**  
Add customizable tags to sticky notes to make it easier to find, review, organize, and categorize important ideas as frames within your mind.

Students below 18yrs cannot apply for any streams.

Students applying for Polytechnic studies should be minimum of 16 yrs.

Students Profile should match with their originals.

If they are PG applicants, then their UG Certificates have to be submitted.

All the required certificates, proofs have to be submitted during registration.

Required Certificates have to be submitted if the applicants is a handicapped person or specially abled.

Certificate proof has to be uploaded if they are First Graduates.

Original certificates of students have to be maintained as Confidential.

Students who are applying for Sports Quota have to submit their Sports certificates.

Marks of entrance exams is considered.

Student's Tenth, Twelfth marks, CGPA for PG applicants have to be collected.

Admissions are done based on the marks obtained by students in University level Entrance Exams.

Proof of entrance exams specific to their applied course has to be submitted.

Based on their scores, branches of University will vary.

According to their suggestions, specially abled people are given location.

Students under reservation quota are given more preference for admission.

The cutoff requirement of admission is done based on caste.

If their profile details are not matched, their admission will be rejected.

If their marks are below the expectation of University, their application will be rejected.

Students from other states receive preference over those with specially abled.

### 3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Generally as the students don't have much idea about the procedures, requirements and details of the universities in the India they seek help from the education consultancy firms to help them successfully secure admission in the universities which are best suitable for their profile, for this they have to invest huge amount of money as consultancy fees.
2.	Idea / Solution description	Providing an as accurate as possible prediction for the student's chances of admissions to the universities of their choice based on their academic transcripts with the help of an Web Application and Machine Learning algorithms
3.	Novelty / Uniqueness	It seems that there are no application or web application which is common for all over India for predicting the eligibility of an student for Indian university, so the proposed application will contain the eligibility criteria of all the Indian universities.
4.	Social Impact / Customer Satisfaction	It helps student for making decision for choosing a right college. Allows students to check ever college eligibility criteria. Avoid middle man and make direct connection between Students and Universities.
5.	Business Model (Revenue Model)	Like most areas of the educational world these days, technology is forcing higher education institutions to do more with less. Institutions are under increasing pressure to admit more students, retain these students, and do their best to ensure student success. Facing this pressure, tech-savvy institutions can benefit greatly from predictive analytics and predictive models to help achieve their goals.
6.	Scalability of the Solution	This will also help you to finalize your dream schools with a realistic road map, with the help of information coupled with a bit of reality check on your academic scores, work experience, your eminence over your peers. we have connoisseurs who shall work with you to amplify your prospects of receiving offers by ensuring that universities you apply.

## 3.4 Problem Solution Fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> Who is your customer? I.e. working parents of 0-5 y.o. kids	CS	<b>6. CUSTOMER CONSTRAINTS</b> What constraints prevent your customers from taking action or limit their choice of solution? I.e. spending power, budget, no cash, network connection, availability	CC	<b>5. AVAILABLE SOLUTIONS</b> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? I.e. pen and paper is an alternative to digital notetaking	AS
	<ul style="list-style-type: none"> <li>Students who wish to know their prospects of being admitted to a specific university;</li> <li>students who have earned their UG degree or diploma; students who have passed their Twelfth Standard.</li> </ul>		<ul style="list-style-type: none"> <li>Online applications make it easier for fraudsters to manipulate the application process and eligibility requirements.</li> <li>Most rural areas experience high blackouts and electricity problems.</li> </ul>		<ul style="list-style-type: none"> <li>The existing ML applications for Admission prediction can be made better and more effective by training with more datasets and better methodologies like logistic regression.</li> </ul>	
Focus on J&P, tap into BE, understand the	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> Which job-to-be-done (or problems) do you address for your customer? could be more than one, explore different sides.	J&P	<b>9. PROBLEM ROOT CAUSE</b> What is the real reason that this problem exists? What is the back story behind the need to do this job? I.e. customers have to do it because of the change in regulations.	RC	<b>7. BEHAVIOUR</b> What does your customer do to address the problem and get the job done? I.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)	BE
	<ul style="list-style-type: none"> <li>The aim of this project is to help students in shortlisting universities with their profiles.</li> <li>The predicted output gives them a fair idea about their admission chances in a particular university.</li> <li>This analysis should also help students who are currently preparing or will be preparing to get a better idea.</li> </ul>		<ul style="list-style-type: none"> <li>Online college admissions are easier for students to complete than offline ones, which take a lot of time.</li> <li>It is challenging to access the students' records physically in offline admissions.</li> </ul>		<ul style="list-style-type: none"> <li>Students provide their academic details to obtain accurate results.</li> <li>Students should explore the available universities.</li> </ul>	
Identify strong TR & EM	<b>3. TRIGGERS</b> What triggers customers to act? I.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.	TR	<b>10. YOUR SOLUTION</b> If you are working on an existing business, write down your current solution first in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.	SL	<b>8. CHANNELS of BEHAVIOUR</b> 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7.	CH
	<ul style="list-style-type: none"> <li>Students would find it very convenient to complete their admissions online because we live in a digital era, and machine learning is also becoming a new trend.</li> </ul>		<ul style="list-style-type: none"> <li>ML-based Application for University Admit Eligibility Predictor.</li> <li>Obtain the marks of the students using which the ML model predicts their chances of admission by evaluating with the help of efficient algorithms like Logistic Regression.</li> </ul>		<ul style="list-style-type: none"> <li>Using the University Predictor online is not always secure and may have the risk of stealing the confidential data of users.</li> </ul>	
	<b>4. EMOTIONS: BEFORE / AFTER</b> How do customers feel when they face a problem or a job and afterwards? I.e. lost, insecure -> confident, in control - use it in your communication strategy & design.	EM			8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.	
	<ul style="list-style-type: none"> <li>ML-based application prevents the students from the trouble of attending career guidance programs.</li> </ul>				<ul style="list-style-type: none"> <li>Ensure the safety and maintenance of their academic records.</li> </ul>	

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Data Collection	He following details of Students' Score are collected: 1.HSC 2.SSLC 3.CGPA if their PG Applicants.
FR-4	Evaluation	Using ML algorithms to analyse the data entered by the students and testing the developed ML model with the supplied data.
FR-5	Prediction	Prediction is done based on the result of evaluation, the List of Universities for which the students are eligible to apply will be displayed.
FR-6	Output	Based on their eligibility, students move forward with the admissions procedure to the predicted university and course.

### 4.2 Non-functional Requirements:

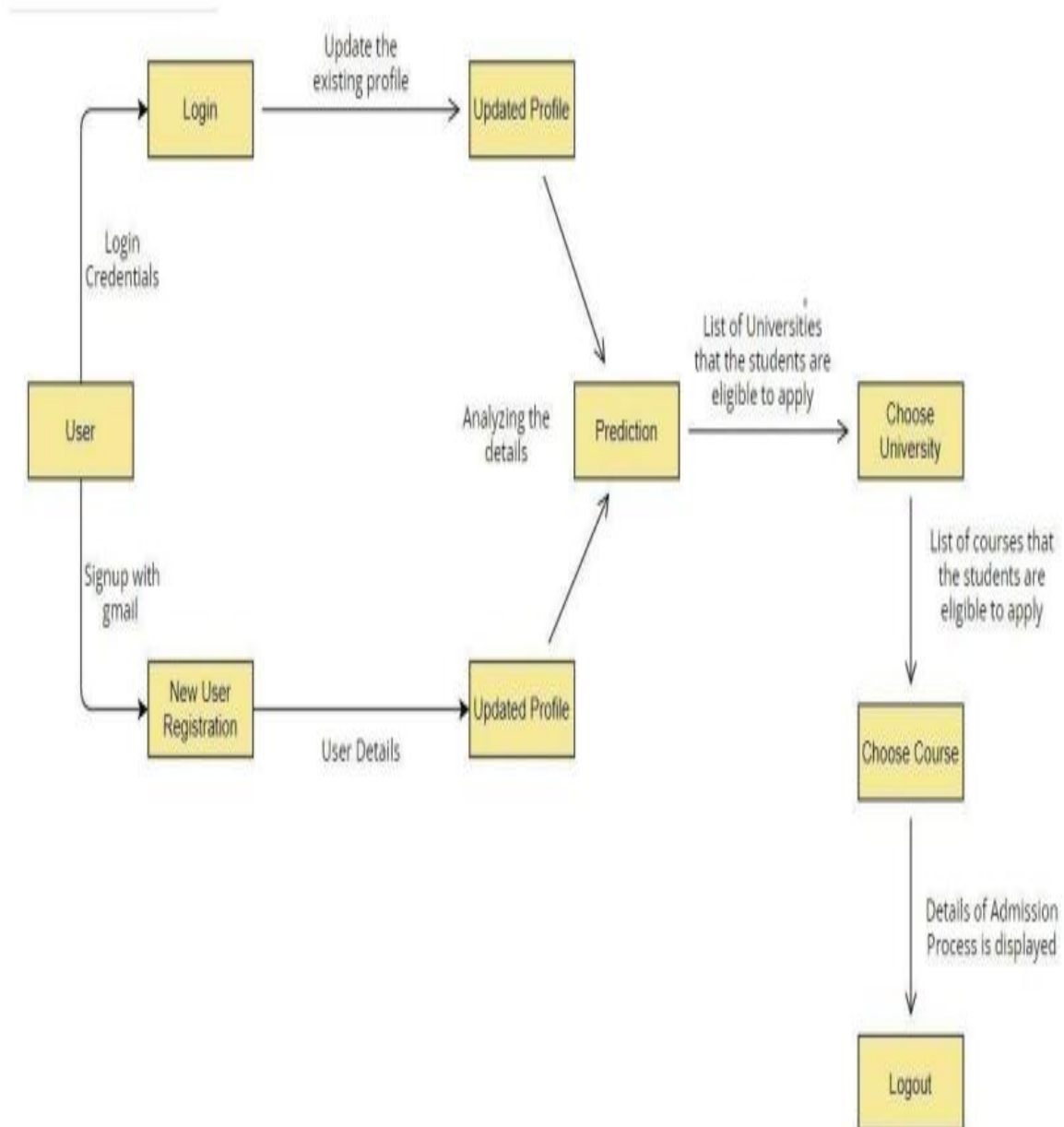
Following are the non-functional requirements of the proposed solution.

<b>NFR No.</b>	<b>Non-Functional Requirements</b>	<b>Description</b>
NFR-1	Usability	Interactive and Effective UI Visualization of Progress Customer Satisfaction Ease of Learning
NFR-2	Security	Frequent Updates using the Customers' feedback. Automatic Logout when the app is not in use to prevent unauthorized access to the user's account.
NFR-3	Reliability	The predictor system will be consistent in order for the system to produce trustworthy and accurate outcomes.
NFR-4	Performance	As logistic regression is applied to develop, performance will be more effective.
NFR-5	Availability	Users will be able to access the system predictor at any time, anyplace, as needed.
NFR-6	Scalability	It can handle any amount of data and perform many computations in a cost - effective and time - saving way.

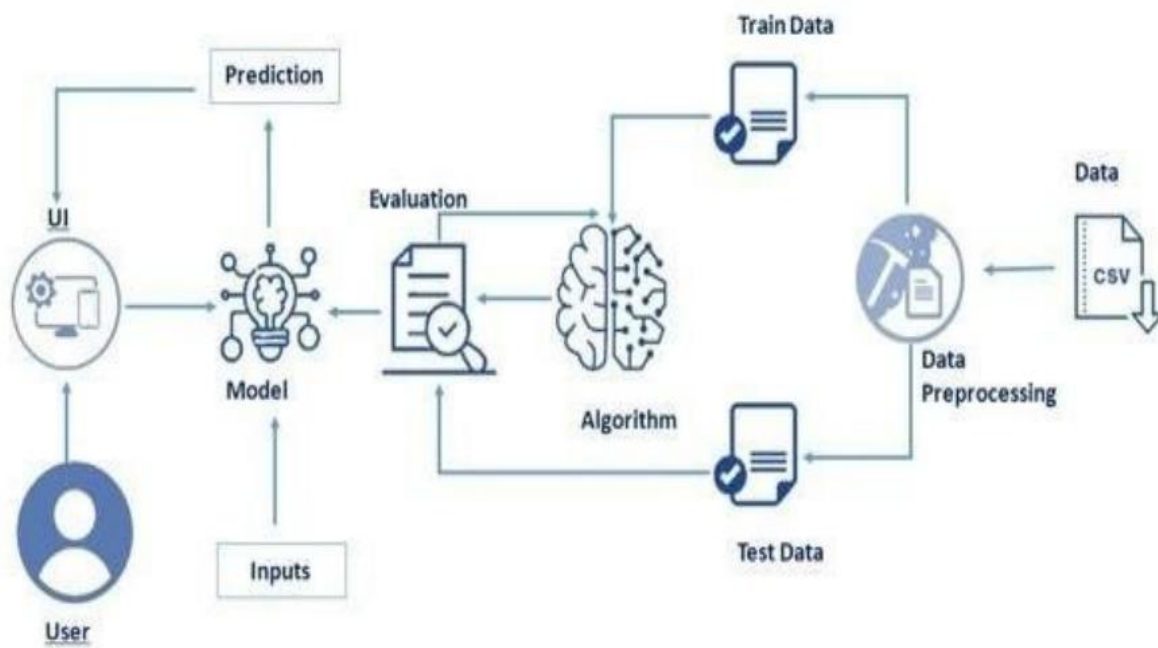


## 5. PROJECT DESIGN












### 5.1 Data Flow Diagram



## 5.2 Solution & Technical Architecture

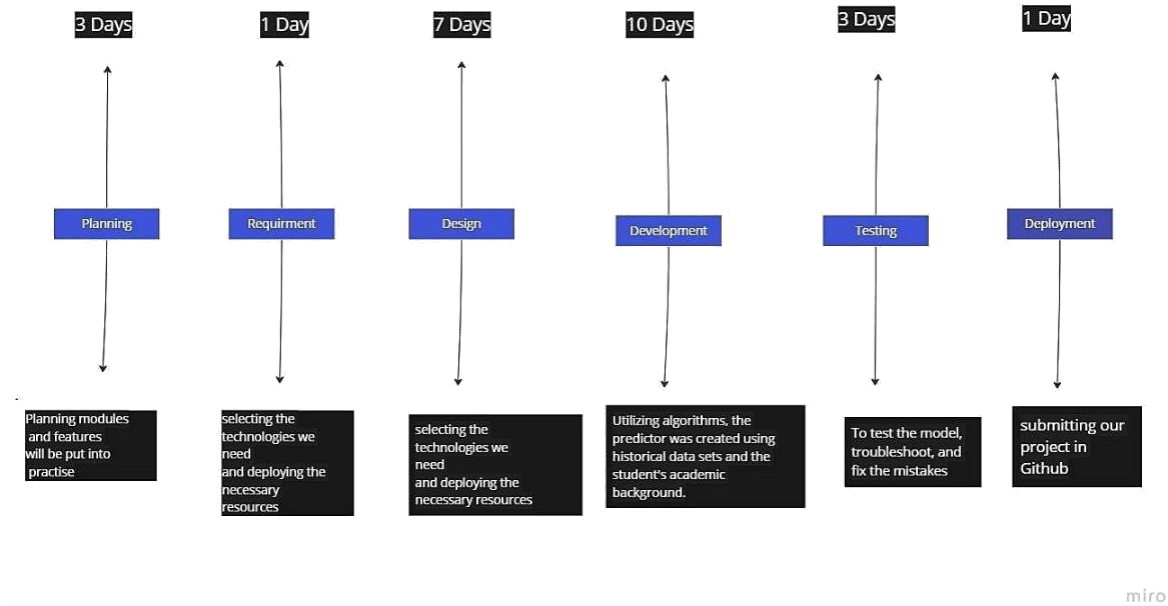


## 5.3 User Stories

		 <b>Entice</b> How does someone initially become aware of this process?	 <b>Enter</b> What do people experience as they begin the process?	 <b>Engage</b> In the core moments in the process, what happens?	 <b>Exit</b> What do people typically experience as the process finishes?	 <b>Extend</b> What happens after the experience is over?
 <b>Steps</b> What does the person (or group) typically experience?		<b>Generate and Remembrance</b> Typical things that someone might experience when they first become aware of the process.	<b>System Login</b> Typical things that someone might experience when they first log in to the system.	<b>Initiation</b> Typical things that someone might experience when they first start the process.	<b>Adaptation Phase</b> Typical things that someone might experience when they first adapt to the process.	<b>Feedback</b> Typical things that someone might experience when they first give feedback.
 <b>Interactions</b> What interactions do they have at each step along the way? • <b>People:</b> Who do they see or talk to? • <b>Places:</b> Where are they? • <b>Things:</b> What digital touchpoints or physical objects would they use?		Typical things that someone might experience when they first become aware of the process.	Typical things that someone might experience when they first log in to the system.	Typical things that someone might experience when they first start the process.	Typical things that someone might experience when they first adapt to the process.	Typical things that someone might experience when they first give feedback.
 <b>Goals &amp; motivations</b> At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")		Typical things that someone might experience when they first become aware of the process.	Typical things that someone might experience when they first log in to the system.	Typical things that someone might experience when they first start the process.	Typical things that someone might experience when they first adapt to the process.	Typical things that someone might experience when they first give feedback.
 <b>Positive moments</b> What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?		Typical things that someone might experience when they first become aware of the process.	Typical things that someone might experience when they first log in to the system.	Typical things that someone might experience when they first start the process.	Typical things that someone might experience when they first adapt to the process.	Typical things that someone might experience when they first give feedback.
 <b>Negative moments</b> What steps does a typical person find frustrating, confusing, engaging, costly, or time-consuming?		Typical things that someone might experience when they first become aware of the process.	Typical things that someone might experience when they first log in to the system.	Typical things that someone might experience when they first start the process.	Typical things that someone might experience when they first adapt to the process.	Typical things that someone might experience when they first give feedback.
 <b>Areas of opportunity</b> How might we make each step better? What ideas do we have? What have others suggested?		Typical things that someone might experience when they first become aware of the process.	Typical things that someone might experience when they first log in to the system.	Typical things that someone might experience when they first start the process.	Typical things that someone might experience when they first adapt to the process.	Typical things that someone might experience when they first give feedback.

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation



### 6.2 Sprint Delivery Schedule

Sprint	Functiona l Requirem ent	User Story No.	User Story / Task	Story Point s	Priori ty	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	2	High	DEVADHARSHINI M
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	DHARANI V
Sprint-2		USN-3	As a user, I can register for the application	2	Low	DIVYA J

			through Facebook			
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	DIVYA BHARATHI S
Sprint-1	login	USN-5	As a user, I can log into the application by entering email & password	1	High	DEVA DHARSHINI M
	dashboard					

Sprint	Total Story points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as onPlanned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Oct 2022	20	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Oct 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Oct 2022	20	14 Nov 2022

## 7. CODING & SOLUTIONING

### 7.1 Feature

The new feature will predict the chances in the admission of the university. The feature was designed in the html code connected with app.py as the backend.

```
<!DOCTYPE html>
```

```
0hdf<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="/css/styles.css">
<title>University Admit Eligibility Predictor</title>
</head>
<body class="home">
  <h1>University Admit Eligibility Predictor</h1>
  <form method="post" action="/predict" class="form">
    <div class="form-data">
      <label for="greScore">GRE Score:</label>
      <input name="greScore" placeholder="GRE Score" required>
    </div>
    <div class="form-data">
      <label for="toeflScore">TOEFL Score:</label>
      <input name="toeflScore" placeholder="TOEFL Score" required>
    </div>
    <div class="form-data">
      <label for="univRank">University Rank:</label>
      <input name="univRank" placeholder="University Rank" required>
    </div>
    <div class="form-data">
      <label for="sop">SOP:</label>
      <input name="sop" placeholder="SOP" required>
    </div>
    <div class="form-data">
      <label for="lor">LOR:</label>
      <input name="lor" placeholder="LOR" required>
    </div>
```

```

<div class="form-data">
    <label for="cgpa">CGPA:</label>
    <input name="cgpa" placeholder="CGPA" required>
</div>
<div class="form-data">
    <label for="research">Research:</label>
    <input name="research" placeholder="Research" required>
</div>
<br>
<br>
<button type="submit" class="btns">Submit</button>
<a href="/"><button type="button" class="btns">Back</button></a>
</form>
</body>
</html>

```

## 7.3 Database Schema

The database used here in this project was Admission\_Predict.csv. The sample screenshot of the database are,

C:\Users\GABE\Documents\12.6.18									
Search this file...									
Sr	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOS	CGPA	Research	Chance of Admit
1	1	327	118	4	4.5	4.5	9.65	1	0.92
2	2	324	117	4	4	4.5	9.37	1	0.76
3	3	310	104	3	3	3.5	8	1	0.72
4	4	322	110	3	3.5	2.5	9.47	1	0.8
5	5	314	103	2	2	5	9.25	0	0.45
6	6	300	105	3	4.5	5	9.34	1	0.8
7	7	321	109	3	5	4	9.2	1	0.75
8	8	306	101	2	2	4	7.9	0	0.65
9	9	282	102	1	2	1.5	8	0	0.5
10	10	323	108	3	3.5	3	9.6	0	0.45
11	11	325	100	3	3.5	4	9.4	1	0.33
12	12	327	111	4	4	4.5	9	1	0.84
13	13	328	112	4	4	4.5	9.1	1	0.76
14	14	307	109	3	4	0	8	1	0.63
15	15	311	104	3	2.5	2	9.3	1	0.61
16	16	294	105	3	3.5	2.5	8.3	0	0.54
17	17	317	107	3	4	3	8.7	0	0.69
18	18	290	100	3	4	5	9	1	0.65
19	19	298	100	3	4	5	8.8	0	0.63
20	20	309	102	3	3.5	5	9.5	0	0.62
21	21	312	101	3	3	2	7.9	1	0.66
22	22	325	114	4	2	2	9.4	0	0.7
23	23	300	106	5	5	5	9.5	1	0.84
24	24	304	109	3	5	4.5	9.7	1	0.85
25	25	290	109	3	4	3.5	9.8	1	0.97

## **8. TESTING**

### **User Acceptance Testing**

- **Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the University Admit Eligibility Predictor project at the time of the release to User Acceptance Testing (UAT).

- **Defect Analysis**

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved



Solution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	19
Duplicate	0	0	0	0	0
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduce	0	0	0	0	0
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	24	14	13	26	64

This report shows the number of test cases that have passed, failed, and untested

## 9. RESULTS

### 9.1 Performance Metrics

Measure the performance using Metrics

```
pd.crosstab(Y_Test,y_predict)
```

```
col_0    0    1    2
Sex
0  108   29  112
1   33  223   35
2  123   52  121
```

```
print(classification_report(Y_Test,y_predict))
```

```

              precision    recall  f1-score   support

     0       0.41         0.43         0.42         249
     1       0.73         0.77         0.75         291
     2       0.45         0.41         0.43         296

 accuracy          0.54         0.54         0.54         836
 macro avg         0.53         0.54         0.53         836
 weighted avg         0.54         0.54         0.54         836
```

## **10. ADVANTAGES&DISADVANTAGES 10.1**

### **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.
- It is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data.

### **10.2 Dis-Advantages**

- Required active internet connection.
- System will provide inaccurate results if data entered incorrectly.

## 11. CONCLUSION

This system ,being the first we have created in Python using ML algorithms and other front end languages such as html, css, java script , has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself. Debugging can sometimes be a relatively straight forward process, or rather finding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first. Language used must be simple and easy to understand and compatibility is paramount. If this system were not designed as an entirely web based application, it would not have been possible to recreate its current state of portability. Overall, the system performs well, and while it does not include all of the features that may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

## 12. FUTURE SCOPE

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.

## 13. APPENDIX

### 13.1 Source Code

#### PYTHON CODE

Uploading the python code ,

#### Import Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

#### Data Collection

In [16]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.# You might want to remove those credentials before you share the
notebook.cos_client = ibm_boto3.client(service_name='s3',
```

```

ibm_api_key_id='IYMIjb2yXoDD9i4rh2MgnyvRiI_9CoyB3MeIAbEk4JMK',
ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
config=Config(signature_version='oauth'),
endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
bucket = 'ibmuniv-donotdelete-pr-xvu5spl8e33fmm'object_key = 'Admission_Predict.csv'
body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']# add missing __iter__
method, so pandas accepts body as file-like objectif not hasattr(body, "__iter__"):
body.__iter__ = types.MethodType( __iter__, body )
data = pd.read_csv(body)data.head()

```

Out[16]:

	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 [41]:

```
data.describe()
```

Out[41]:

	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

In [17]:

```
data.info()
```

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

## Handling Missing Values

In [18]:

```
data.isnull().any()
```

Out[18]:

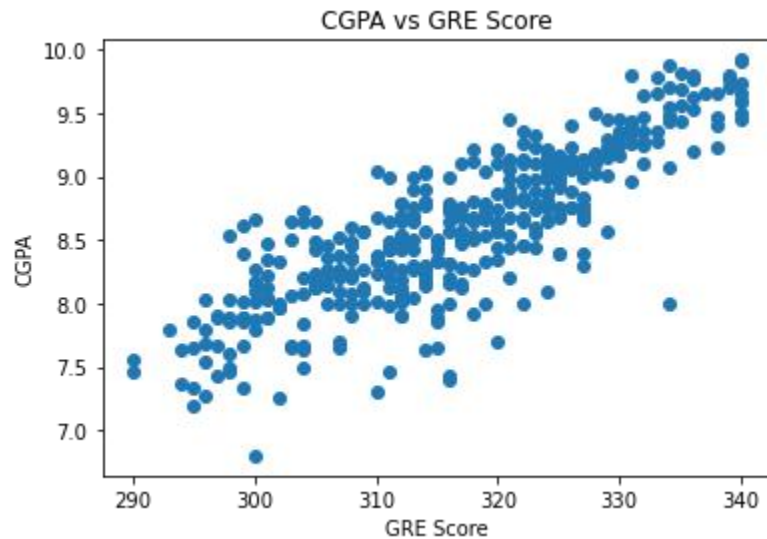
Serial No.	False
GRE Score	False
TOEFL Score	False
University Rating	False
SOP	False
LOR	False
CGPA	False
Research	False
Chance of Admit	False

dtype: bool

## Data Visualization

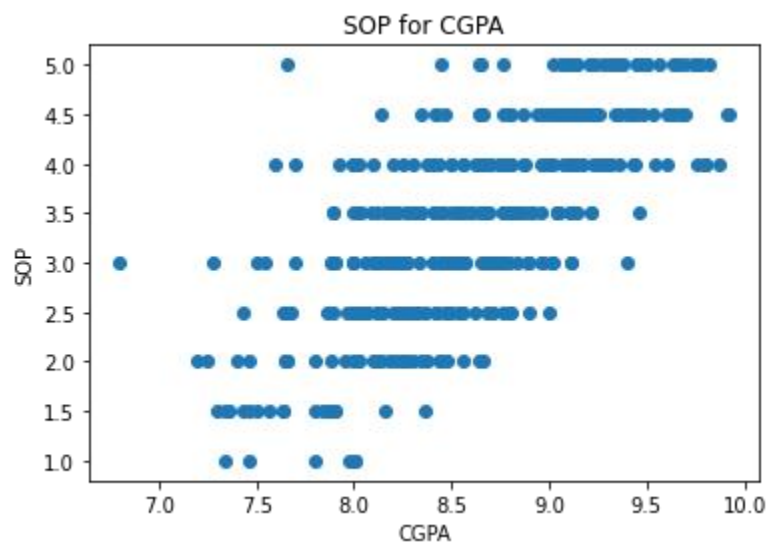
In [19]:

```
plt.scatter(data['GRE Score'],data['CGPA'])plt.title('CGPA vs GRE Score')plt.xlabel('GRE Score')plt.ylabel('CGPA')plt.show()
```



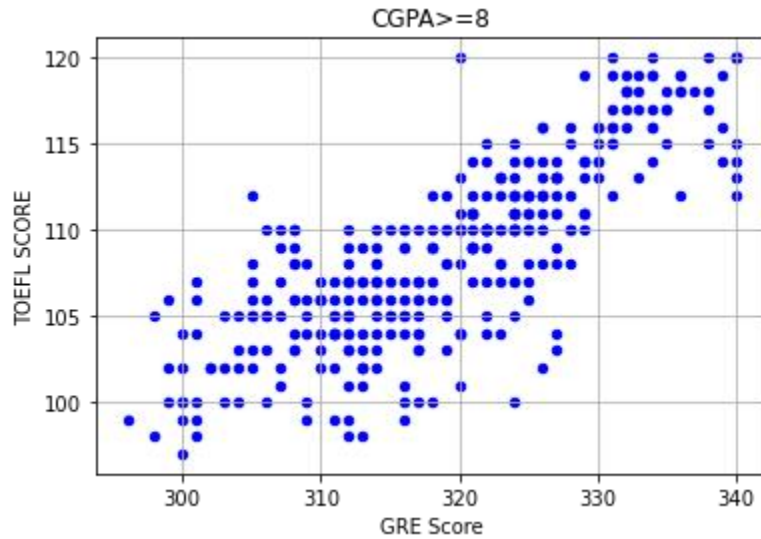
In [42]:

```
plt.scatter(data['CGPA'],data['SOP'])plt.title('SOP for
CGPA')plt.xlabel('CGPA')plt.ylabel('SOP')plt.show()
```



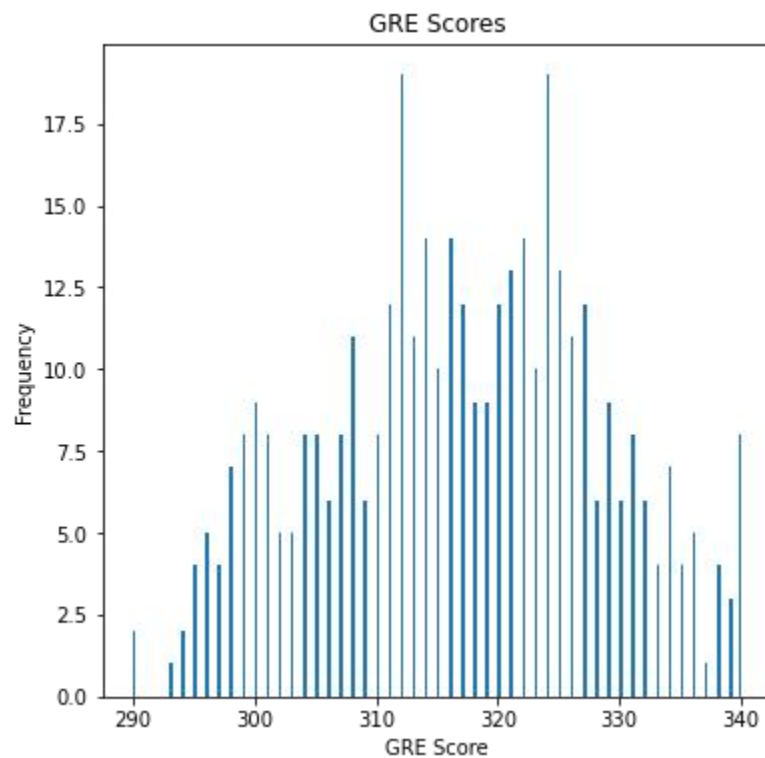
In [43]:

```
data[data.CGPA >= 8].plot(kind='scatter', x='GRE Score', y='TOEFL
Score',color="BLUE")plt.xlabel("GRE Score")plt.ylabel("TOEFL
SCORE")plt.title("CGPA>=8")plt.grid(True)plt.show()
```



In [44]:

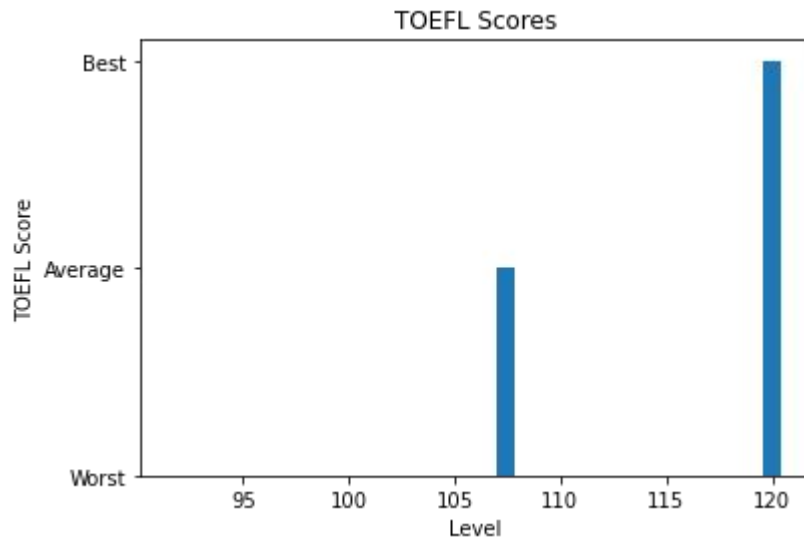
```
data["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))plt.title("GRE Scores")plt.xlabel("GRE Score")plt.ylabel("Frequency")plt.show()
```



In [45]:

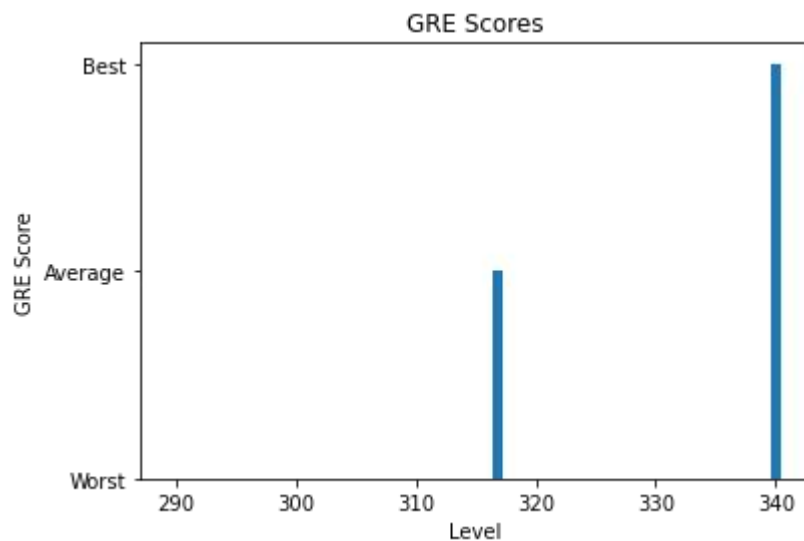
```
p = np.array([data["TOEFL Score"].min(),data["TOEFL Score"].mean(),data["TOEFL Score"].max()])r = ["Worst","Average","Best"]plt.bar(p,r)plt.title("TOEFL Scores")plt.xlabel("Level")plt.ylabel("TOEFL Score")plt.show()
```





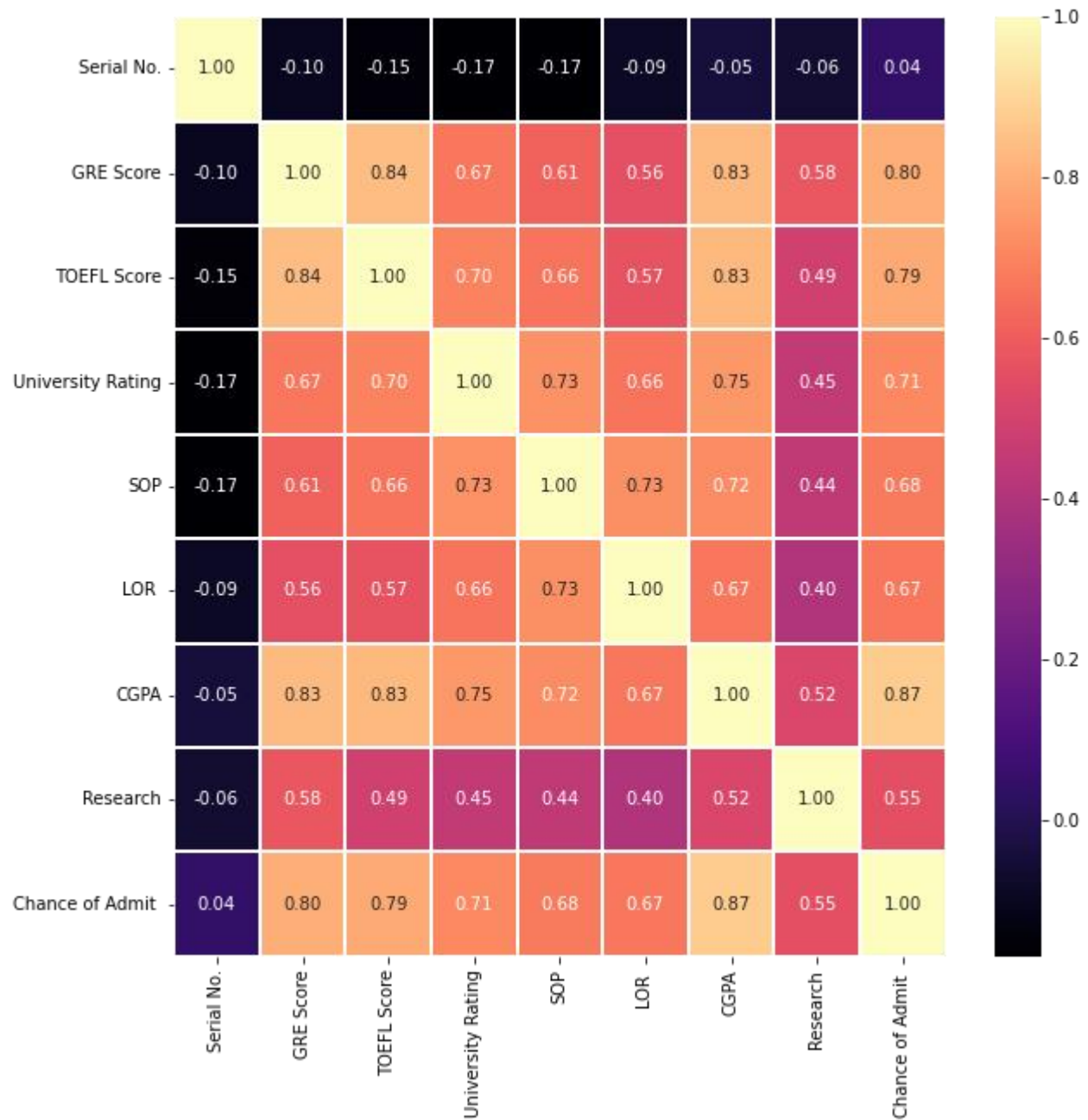
In [46]:

```
g = np.array([data["GRE Score"].min(),data["GRE Score"].mean(),data["GRE Score"].max()])h = ["Worst","Average","Best"]plt.bar(g,h)plt.title("GRE Scores")plt.xlabel("Level")plt.ylabel("GRE Score")plt.show()
```



In [47]:

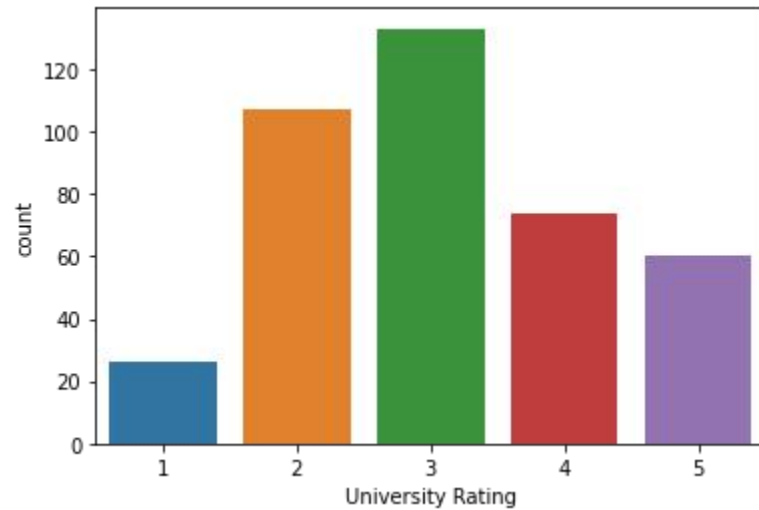
```
plt.figure(figsize=(10, 10))sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt='.2f',cmap="magma")plt.show()
```



In [48]:

```
data.Research.value_counts()sns.countplot(x="University Rating",data=data)
```

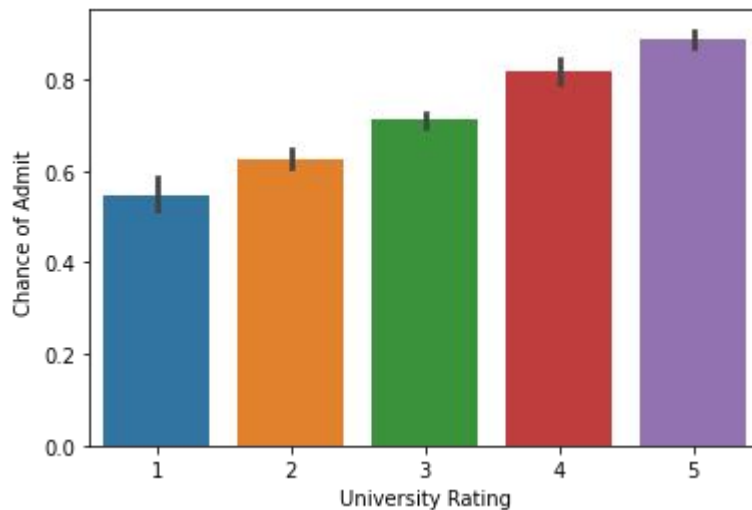
Out[48]:



In [49]:

```
sns.barplot(x="University Rating", y="Chance of Admit ", data=data)
```

Out[49]:



## Splitting Dependent and Independent Columns

In [20]:

```
x = data.iloc[:, :-1]y = data['Chance of Admit ']
```

In [21]:

```
x.head()
```

Out[21]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	1	337	118	4	4.5	4.5	9.65	1
1	2	324	107	4	4.0	4.5	8.87	1
2	3	316	104	3	3.0	3.5	8.00	1
3	4	322	110	3	3.5	2.5	8.67	1

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
4	5	314	103	2	2.0	3.0	8.21	0

In [22]:

```
y.head()
```

Out[22]:

```
0    0.92
1    0.76
2    0.72
3    0.80
4    0.65
```

Name: Chance of Admit , dtype: float64

## Splitting the Data into Train and Test

In [23]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=50)
```

## Model Building

In [24]:

```
from sklearn.ensemble import GradientBoostingRegressor
model = GradientBoostingRegressor().fit(x_train, y_train)
```

Out[24]:

```
GradientBoostingRegressor()
```

In [25]:

```
model.score(x_test, y_test)
```

Out[25]:

```
0.8045553566319525
```

In [26]:

```
from sklearn.metrics import accuracy_score
y_predict = model.predict(x_test)
```

In [27]:

```
y_train = (y_train > 0.5)
y_test = (y_test > 0.5)
```

In [28]:

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(random_state=0, max_iter=1000)
lr_model = lr.fit(x_train, y_train)
y_pred = lr_model.predict(x_test)
```

## Model Evaluation

In [29]:

```
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix
```

```
print('Accuracy Score:', accuracy_score(y_test, y_pred)) print('Recall Score:',  
recall_score(y_test, y_pred)) print('ROC AUC Score:', roc_auc_score(y_test,  
y_pred))print('Confusion Matrix:\n', confusion_matrix(y_test, y_pred))
```

Accuracy Score: 0.9375

Recall Score: 0.9487179487179487

ROC AUC Score: 0.7243589743589743

Confusion Matrix:

```
[[ 1  1]  
[ 4 74]]
```

## Save the Model

In [50]:

```
# import pickle  
# pickle.dump(lr, open("university.pkl", 'wb'))# model = pickle.load(open("university.pkl", 'rb'))
```

## IBM Deployment

In [30]:

```
!pip install -U ibm-watson-machine-learning
```

Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)

Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)

Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)

Requirement already satisfied: ibm-cos-sdk==2.11.\* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)

Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.3.4)

Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)

Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)

Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)

Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)

Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.11.0)

Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (0.10.0)

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.8.2)

Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)

Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)

Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (1.15.0)

Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.3)

Requirement already satisfied: charset-normalizer~2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (2.0.4)

Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm-watson-machine-learning) (3.6.0)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-learning) (3.0.4)

In [31]:

```
from ibm_watson_machine_learning import APIClientimport json
```

In [32]:

```
wml_credentials = {  
    "apikey": "2P76TOf0kpxjsnHwF-xwPOLMqVBBviAH0TPSUWUnUb4M",  
    "url": "https://us-south.ml.cloud.ibm.com"}
```

In [33]:

```
wml_client = APIClient(wml_credentials)wml_client.spaces.list()
```

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceeded 50

```
-----  
ID              NAME  CREATED  
1abc7d7a-f529-4419-9afd-b5de22df88be models 2022-11-16T11:23:56.373Z  
-----
```

In [34]:

```
SPACE_ID = "1abc7d7a-f529-4419-9afd-  
b5de22df88be"wml_client.set.default_space(SPACE_ID)
```

Out[34]:

'SUCCESS'

In [35]:

```
wml_client.software_specifications.list(500)
```

```
-----
NAME                ASSET_ID                TYPE
default_py3.6       0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base
kernel-spark3.2-scala2.12  020d69ce-7ac1-5e68-ac1a-31189867356a base
pytorch-onnx_1.3-py3.7-edt  069ea134-3346-5748-b513-49120e15d288 base
scikit-learn_0.20-py3.6    09c5a1d0-9c1e-4473-a344-eb7b665ff687 base
spark-mllib_3.0-scala_2.12  09f4cff0-90a7-5899-b9ed-1ef348aebdee base
pytorch-onnx_rt22.1-py3.9   0b848dd4-e681-5599-be41-b5f6fccc6471 base
ai-function_0.1-py3.6      0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base
shiny-r3.6             0e6e79df-875e-4f24-8ae9-62dcc2148306 base
tensorflow_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base
pytorch_1.1-py3.6        10ac12d6-6b30-4ccd-8392-3e922c096a92 base
tensorflow_1.15-py3.6-ddl   111e41b3-de2d-5422-a4d6-bf776828c4b7 base
autoai-kb_rt22.2-py3.10    125b6d9a-5b1f-5e8d-972a-b251688ccf40 base
runtime-22.1-py3.9        12b83a17-24d8-5082-900f-0ab31fbfd3cb base
scikit-learn_0.22-py3.6    154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base
default_r3.6            1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base
pytorch-onnx_1.3-py3.6     1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base
kernel-spark3.3-r3.6       1c9e5454-f216-59dd-a20e-474a5cdf5988 base
pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base
tensorflow_2.1-py3.6       1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base
spark-mllib_3.2          20047f72-0a98-58c7-9ff5-a77b012eb8f5 base
tensorflow_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base
runtime-22.1-py3.9-cuda    26215f05-08c3-5a41-a1b0-da66306ce658 base
do_py3.8               295addb5-9ef9-547e-9bf4-92ae3563e720 base
autoai-ts_3.8-py3.8       2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base
tensorflow_1.15-py3.6     2b73a275-7cbf-420b-a912-eae7f436e0bc base
kernel-spark3.3-py3.9      2b7961e2-e3b1-5a8c-a491-482c8368839a base
pytorch_1.2-py3.6         2c8ef57d-2687-4b7d-acce-01f94976dac1 base
spark-mllib_2.3          2e51f700-bca0-4b0d-88dc-5c6791338875 base
pytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base
spark-mllib_3.0-py37       36507ebe-8770-55ba-ab2a-eafe787600e9 base
spark-mllib_2.4          390d21f8-e58b-4fac-9c55-d7ceda621326 base
autoai-ts_rt22.2-py3.10    396b2e83-0953-5b86-9a55-7ce1628a406f base
xgboost_0.82-py3.6        39e31acd-5f30-41dc-ac44-60233c80306e base
pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base
```

pytorch-onnx_rt22.2-py3.10	40e73f55-783a-5535-b3fa-0c8b94291431	base
default_r36py38	41c247d3-45f8-5a71-b065-8580229facf0	base
autoai-ts_rt22.1-py3.9	4269d26e-07ba-5d40-8f66-2d495b0c71f7	base
autoai-obm_3.0	42b92e18-d9ab-567f-988a-4240ba1ed5f7	base
pmml-3.0_4.3	493bcb95-16f1-5bc5-bee8-81b8af80e9c7	base
spark-mllib_2.4-r_3.6	49403dff-92e9-4c87-a3d7-a42d0021c095	base
xgboost_0.90-py3.6	4ff8d6c2-1343-4c18-85e1-689c965304d3	base
pytorch-onnx_1.1-py3.6	50f95b2a-bc16-43bb-bc94-b0bed208c60b	base
autoai-ts_3.9-py3.8	52c57136-80fa-572e-8728-a5e7cbb42cde	base
spark-mllib_2.4-scala_2.11	55a70f99-7320-4be5-9fb9-9edb5a443af5	base
spark-mllib_3.0	5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9	base
autoai-obm_2.0	5c2e37fa-80b8-5e77-840f-d912469614ee	base
spss-modeler_18.1	5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b	base
cuda-py3.8	5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e	base
autoai-kb_3.1-py3.7	632d4b22-10aa-5180-88f0-f52dfb6444d7	base
pytorch-onnx_1.7-py3.8	634d3cdc-b562-5bf9-a2d4-ea90a478456b	base
spark-mllib_2.3-r_3.6	6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c	base
tensorflow_2.4-py3.7	65e171d7-72d1-55d9-8ebb-f813d620c9bb	base
spss-modeler_18.2	687eddc9-028a-4117-b9dd-e57b36f1efa5	base
pytorch-onnx_1.2-py3.6	692a6a4d-2c4d-45ff-a1ed-b167ee55469a	base
spark-mllib_2.3-scala_2.11	7963efe5-bbec-417e-92cf-0574e21b4e8d	base
spark-mllib_2.4-py37	7abc992b-b685-532b-a122-a396a3cdbaab	base
caffe_1.0-py3.6	7bb3dbe2-da6e-4145-918d-b6d84aa93b6b	base
pytorch-onnx_1.7-py3.7	812c6631-42b7-5613-982b-02098e6c909c	base
cuda-py3.6	82c79ece-4d12-40e6-8787-a7b9e0f62770	base
tensorflow_1.15-py3.6-horovod	8964680e-d5e4-5bb8-919b-8342c6c0dfd8	base
hybrid_0.1	8c1a58c6-62b5-4dc4-987a-df751c2756b6	base
pytorch-onnx_1.3-py3.7	8d5d8a87-a912-54cf-81ec-3914adaa988d	base
caffe-ibm_1.0-py3.6	8d863266-7927-4d1e-97d7-56a7f4c0a19b	base
spss-modeler_17.1	902d0051-84bd-4af6-ab6b-8f6aa6fdeabb	base
do_12.10	9100fd72-8159-4eb9-8a0b-a87e12eefa36	base
do_py3.7	9447fa8b-2051-4d24-9eef-5acb0e3c59f8	base
spark-mllib_3.0-r_3.6	94bb6052-c837-589d-83f1-f4142f219e32	base
cuda-py3.7-opence	94e9652b-7f2d-59d5-ba5a-23a414ea488f	base
nlp-py3.8	96e60351-99d4-5a1c-9cc0-473ac1b5a864	base
cuda-py3.7	9a44990c-1aa1-4c7d-baf8-c4099011741c	base
hybrid_0.2	9b3f9040-9cee-4ead-8d7a-780600f542f7	base
spark-mllib_3.0-py38	9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418	base



autoai-kb_3.3-py3.7	a545cca3-02df-5c61-9e88-998b09dc79af	base
spark-mllib_3.0-py39	a6082a27-5acc-5163-b02c-6b96916eb5e0	base
runtime-22.1-py3.9-do	a7e7dbf1-1d03-5544-994d-e5ec845ce99a	base
default_py3.8	ab9e1b80-f2ce-592c-a7d2-4f2344f77194	base
tensorflow_rt22.1-py3.9	acd9c798-6974-5d2f-a657-ce06e986df4d	base
kernel-spark3.2-py3.9	ad7033ee-794e-58cf-812e-a95f4b64b207	base
autoai-obm_2.0 with Spark 3.0	af10f35f-69fa-5d66-9bf5-acb58434263a	base
default_py3.7_opence	c2057dd4-f42c-5f77-a02f-72bdbd3282c9	base
tensorflow_2.1-py3.7	c4032338-2a40-500a-beef-b01ab2667e27	base
do_py3.7_opence	cc8f8976-b74a-551a-bb66-6377f8d865b4	base
spark-mllib_3.3	d11f2434-4fc7-58b7-8a62-755da64fdaf8	base
autoai-kb_3.0-py3.6	d139f196-e04b-5d8b-9140-9a10ca1fa91a	base
spark-mllib_3.0-py36	d82546d5-dd78-5fbb-9131-2ec309bc56ed	base
autoai-kb_3.4-py3.8	da9b39c3-758c-5a4f-9cfd-457dd4d8c395	base
kernel-spark3.2-r3.6	db2fe4d6-d641-5d05-9972-73c654c60e0a	base
autoai-kb_rt22.1-py3.9	db6afe93-665f-5910-b117-d879897404d9	base
tensorflow_rt22.1-py3.9-horovod	dda170cc-ca67-5da7-9b7a-cf84c6987fae	base
autoai-ts_1.0-py3.7	deef04f0-0c42-5147-9711-89f9904299db	base
tensorflow_2.1-py3.7-horovod	e384fce5-fdd1-53f8-bc71-11326c9c635f	base
default_py3.7	e4429883-c883-42b6-87a8-f419d64088cd	base
do_22.1	e51999ba-6452-5f1f-8287-17228b88b652	base
autoai-obm_3.2	eae86aab-da30-5229-a6a6-1d0d4e368983	base
tensorflow_rt22.2-py3.10	f65bd165-f057-55de-b5cb-f97cf2c0f393	base
do_20.1	f686cdd9-7904-5f9d-a732-01b0d6b10dc5	base
pytorch-onnx_rt22.2-py3.10-edt	f8a05d07-e7cd-57bb-a10b-23f1d4b837ac	base
scikit-learn_0.19-py3.6	f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9	base
tensorflow_2.4-py3.8	fe185c44-9a99-5425-986b-59bd1d2eda46	base

----- ----

In [36]:

```
import sklearnsklearn.__version__
```

Out[36]:

'1.0.2'

In [37]:

```
MODEL_NAME = 'univ_ad'
```

```
DEPLOYMENT_NAME = 'iris_adm'
```

```
DEMO_MODEL = lr_model
```

In [38]:

```
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

```

model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid}
model_details = wml_client.repository.store_model(
    model = DEMO_MODEL,
    meta_props = model_props,
    training_data = x_train,
    training_target = y_train)

```

model\_details

Out[38]:

```

{'entity': {'hybrid_pipeline_software_specs': [],
'label_column': 'Chance of Admit ',
'schemas': {'input': [{'fields': [{'name': 'Serial No.', 'type': 'int64'},
{'name': 'GRE Score', 'type': 'int64'},
{'name': 'TOEFL Score', 'type': 'int64'},
{'name': 'University Rating', 'type': 'int64'},
{'name': 'SOP', 'type': 'float64'},
{'name': 'LOR ', 'type': 'float64'},
{'name': 'CGPA', 'type': 'float64'},
{'name': 'Research', 'type': 'int64'}]},
'id': '1',
'type': 'struct'}],
'output': []},
'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
'name': 'runtime-22.1-py3.9'},
'type': 'scikit-learn_1.0'},
'metadata': {'created_at': '2022-11-16T17:11:57.291Z',
'id': '0944c716-c6b4-4221-b6b6-893d1d74c506',
'modified_at': '2022-11-16T17:11:59.723Z',
'name': 'univ_ad',
'owner': 'IBMid-667000F3HX',
'resource_key': 'cdd58957-c38c-48ca-9af5-e012d9239d60',
'space_id': '1abc7d7a-f529-4419-9afd-b5de22df88be'},
'system': {'warnings': []}}

```

In [39]:

```
model_id = wml_client.repository.get_model_id(model_details)model_id
```

Out[39]:

```
'0944c716-c6b4-4221-b6b6-893d1d74c506'
```

In [40]:

```
deployment_props = {  
    wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,  
    wml_client.deployments.ConfigurationMetaNames.ONLINE: {}}
```

```
deployment = wml_client.deployments.create(  
    artifact_uid = model_id,  
    meta_props = deployment_props)
```

```
#####  
#####
```

Synchronous deployment creation for uid: '0944c716-c6b4-4221-b6b6-893d1d74c506' started

```
#####  
#####
```

initializing

Note: online\_url is deprecated and will be removed in a future release. Use serving\_urls instead.

ready

```
-----  
Successfully finished deployment creation, deployment_uid='363f2f17-a1e5-4b6b-93d1-a79ae7b  
04882'  
-----
```

## HTML CODES

Uploading Html codes

# Chance.html

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/styles.css">
  <title>University Admit Eligibility Predictor</title>
</head>
<body class="predict">
  <div class="prediction-box">
    <h1>Predicting Chance of Admission</h1>
    <div class="prediction">
      <span>Prediction:</span>
      <h3 style="font-weight: 600;">You have a chance</h3>
    </div>
    <a href="/checkEligibility"><button type="button"
class="btns">Back</button></a>
  </div>
</body>
</html>
```

## Demo2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="/css/styles.css">
  <title>University Admit Eligibility Predictor</title>
</head>
<body class="home">
  <h1>University Admit Eligibility Predictor</h1>
  <form method="post" action="/predict" class="form">
    <div class="form-data">
      <label for="greScore">GRE Score:</label>
      <input name="greScore" placeholder="GRE Score" required>
    </div>
    <div class="form-data">
      <label for="toeflScore">TOEFL Score:</label>
```

```
        <input name="toeflScore" placeholder="TOEFL Score"
required>
    </div>
    <div class="form-data">
        <label for="univRank">University Rank:</label>
        <input name="univRank" placeholder="University Rank"
required>
    </div>
    <div class="form-data">
        <label for="sop">SOP:</label>
        <input name="sop" placeholder="SOP" required>
    </div>
    <div class="form-data">
        <label for="lor">LOR:</label>
        <input name="lor" placeholder="LOR" required>
    </div>
    <div class="form-data">
        <label for="cgpa">CGPA:</label>
        <input name="cgpa" placeholder="CGPA" required>
    </div>
    <div class="form-data">
        <label for="research">Research:</label>
        <input name="research" placeholder="Research" required>
```

```
</div>
<br>
<br>
<button type="submit" class="btns">Submit</button>
<a href="/"><button type="button"
class="btns">Back</button></a>
</form>
</body>
</html>
```

## **Index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="/css/styles.css">
  <title>University Admit Eligibility Predictor</title>
</head>
```

```
<body class="index-page">
  <div class="index">
    <h1>University Admit Eligibility Predictor</h1>
    <a href="/checkEligibility"><button type="button"
class="btns">Check Eligibility</button></a>
  </div>
</body>
</html>
```

## Nochance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="/css/styles.css">
  <title>University Admit Eligibility Predictor</title>
</head>
<body class="index-page">
  <div class="index">
```



```
<h1>University Admit Eligibility Predictor</h1>
  <a href="/checkEligibility"><button type="button"
class="btns">Check Eligibility</button></a>
</div>
</body>
</html>
```

## Script.js

```
const button =
document.getElementById('button');
const theForm =
document.getElementById('theForm');
const loading =
document.getElementById('spinner');

const disableButton = () => {

  console.log('Submitting form...');

  button.disabled = true;

  button.className = "btn btn-outline-primary";

  button.innerHTML = "Predicting..."

  loading.style.display = "block"

};
```

```
const enableButton = () => {  
  
  console.log('Loading window...');  
  
  button.disabled = false;
```

```
button.className = "btn btn-primary"

button.innerHTML = "Predict"

loading.style.display = "none"

}
```

```
theForm.onsubmit = disableButton;

window.onload = enableButton;
```

## **Styles.css**

```
*{
    margin: 0;
    box-sizing: border-box;
}

.index-page{
    background-image: url(/img/bg.jpeg);
}

.index{
    display: flex;
```

```
flex-direction: column;
width: fit-content;
height: 20%;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
position: absolute;
padding: 20px;
border-radius: 50px;
background: rgba(255, 255, 255, 0.7);
align-items: center;
justify-content: center;
box-shadow: 2px 2px 20px rgb(0, 0, 0);
}
```

```
.home{
  background-image: url(/img/bg2.jpeg);
}
```

```
.home h1 {
  width: 100%;
  text-align: center;
  margin-top: 5px;
}
```

```
.form{
  display: flex;
```

```
top: 50%;  
left: 50%;  
position: absolute;  
transform: translate(-50%, -50%);  
flex-direction: column;  
width: fit-content;  
height: fit-content;  
padding: 20px;  
border-radius: 50px;  
background: rgba(255, 255, 255, 0.8);  
align-items: center;  
justify-content: center;  
box-shadow: 2px 2px 20px rgb(0, 0, 0);  
}
```

```
.form input{  
width: 400px;  
height: 50px;  
margin-right: 10px;  
margin-block: 20px;  
font-size: large;  
border-radius: 20px;  
border: none;  
padding: 12px;  
box-shadow: 1px 1px 5px rgb(0, 0, 0);  
background-color: rgba(255, 255, 255, 0.8);  
color: #000;
```

```
}
```

```
.form-data{  
    display: flex;  
    flex-direction: row;  
}
```

```
.form-data label{  
    width: 150px;  
    height: 50px;  
    margin-left: 10px;  
    margin-block: 20px;  
    align-items: center;  
    justify-content: center;  
    font-size: large;  
    padding: 12px;  
    color: #000;  
}
```

```
.predict{  
    background-image: url(/img/bg1.jpeg);  
    background-size: cover;  
}
```

```
.prediction-box{  
    background-color: #fff;  
    width: fit-content;
```

```
height: fit-content;
padding: 20px;
display: flex;
flex-direction: column;
top: 50%;
left: 30%;
position: absolute;
transform: translate(-50%, -30%);
border-radius: 30px;
background: rgba(255, 255, 255, 0.9);
box-shadow: 2px 2px 20px rgb(0, 0, 0);
}
```

```
.prediction{
  display: flex;
  flex-direction: row;
  align-items: center;
  padding: 20px;
  margin-block: 20px;
}
```

```
.prediction h3{
  text-decoration: underline;
}
```

```
.btns{
  width: 200px;
```

```
height: 50px;  
font-size: larger;  
font-weight: 500;  
border-radius: 20px;  
border: none;  
margin-top: 20px;  
background-color: #fff;  
transition: 200ms ease-in-out;  
cursor: pointer;  
}
```

```
.btns:hover{  
background-color: #000;  
color: #fff;  
transform: scale(1.1);  
}
```



## PYTHON CODE

### App.py

```
from flask import Flask, render_template, request
import requests

API_KEY =
"dAkQTmsJ7sfRzutZ8fTcNbHZvKD_ZyoxqjtYF7h8VwC7"
token_response =
requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

app = Flask(__name__, static_url_path="")
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/checkEligibility')
```

```

def checkEligibility():
    return render_template('Demo2.html')

@app.route('/predict', methods=['POST'])
def predict():
    greScore = int(request.form['greScore'])
    toeflScore = int(request.form['toeflScore'])
    univRank = int(request.form['univRank'])
    sop = float(request.form['sop'])
    lor = float(request.form['lor'])
    cgpa = float(request.form['cgpa'])
    research = int(request.form['research'])
    array_of_input_fields = ['greScore', 'toeflScore', 'univRank', 'sop', 'lor',
                             'cgpa', 'research']
    array_of_values_to_be_scored = [greScore, toeflScore, univRank, sop,
    lor, cgpa, research]
    payload_scoring = {"input_data": [{"fields": [array_of_input_fields],
    "values": [array_of_values_to_be_scored]}]}
    response_scoring = requests.post('https://us-
    south.ml.cloud.ibm.com/ml/v4/deployments/9f4939ed-7f21-4881-8ae4-
    234e7515f65a/predictions?version=2022-10-21', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    predictions = response_scoring.json()

```

```
prediction = predictions['predictions'][0]['values'][0][0]
```

```
if prediction:
```

```
    return render_template('chance.html')
```

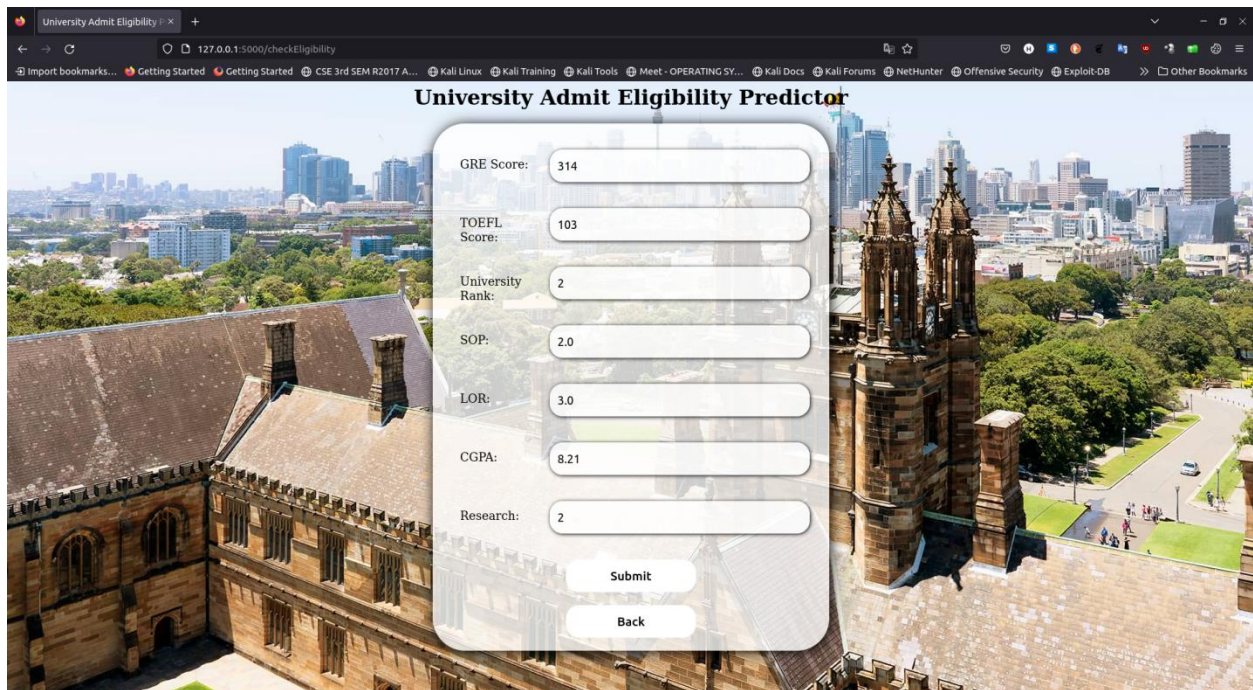
```
else:
```

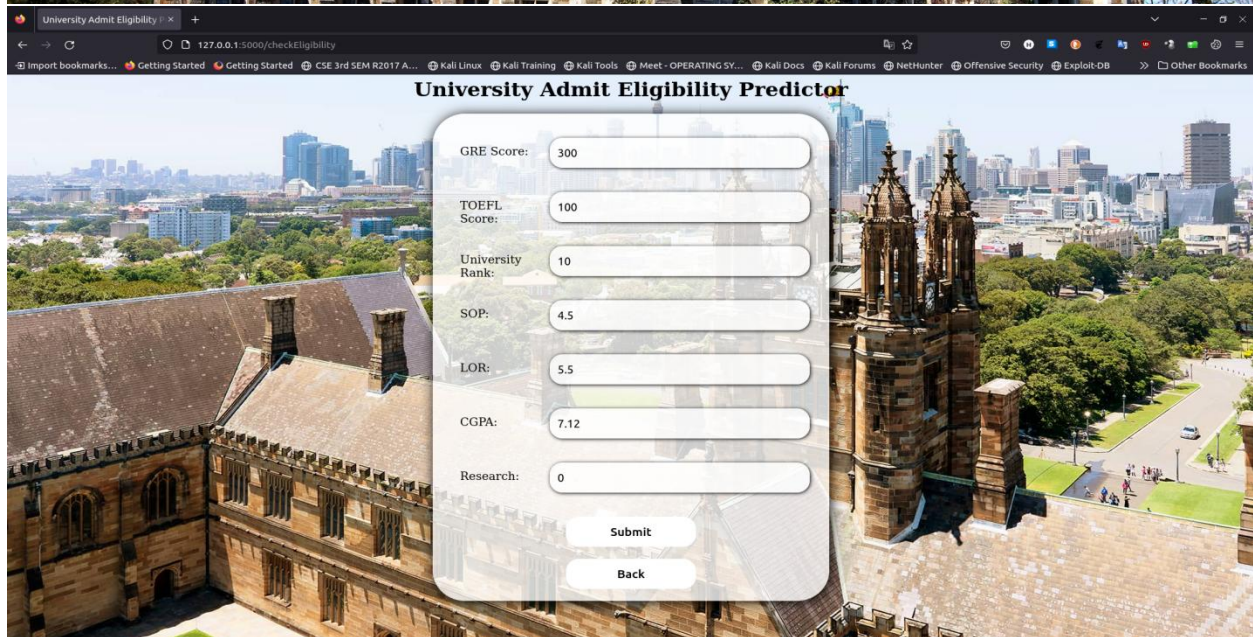
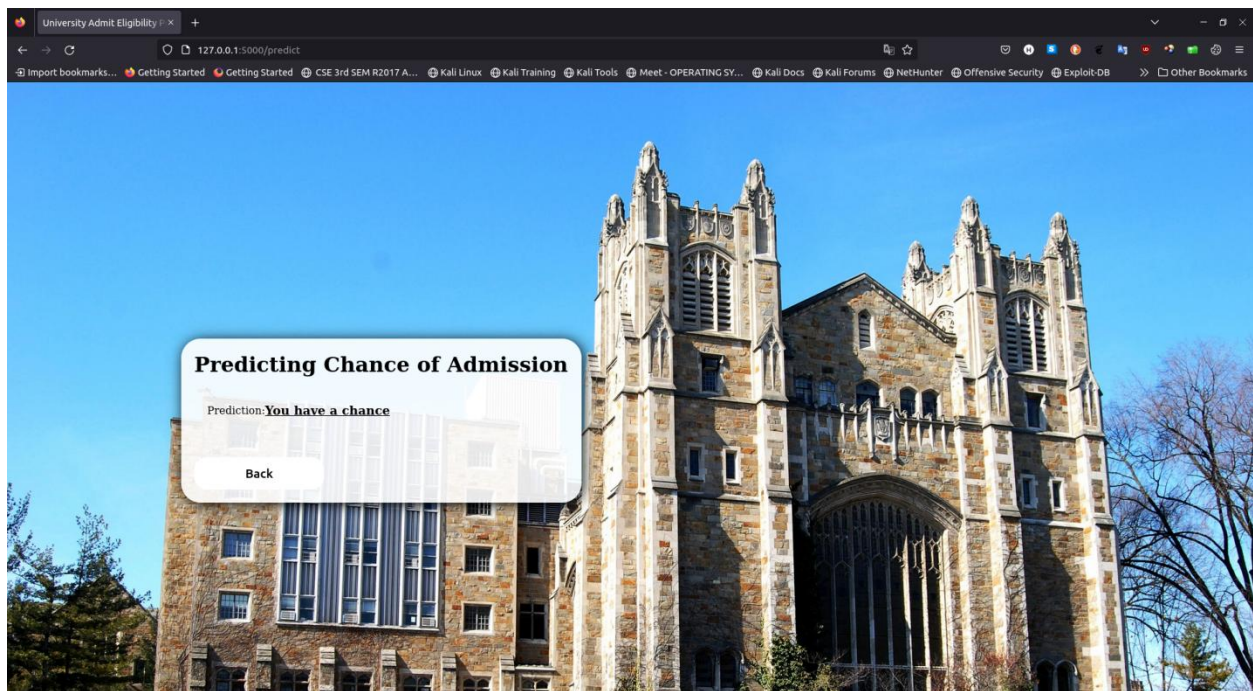
```
    return render_template('noChance.html')
```

```
if __name__ == "__main__":
```

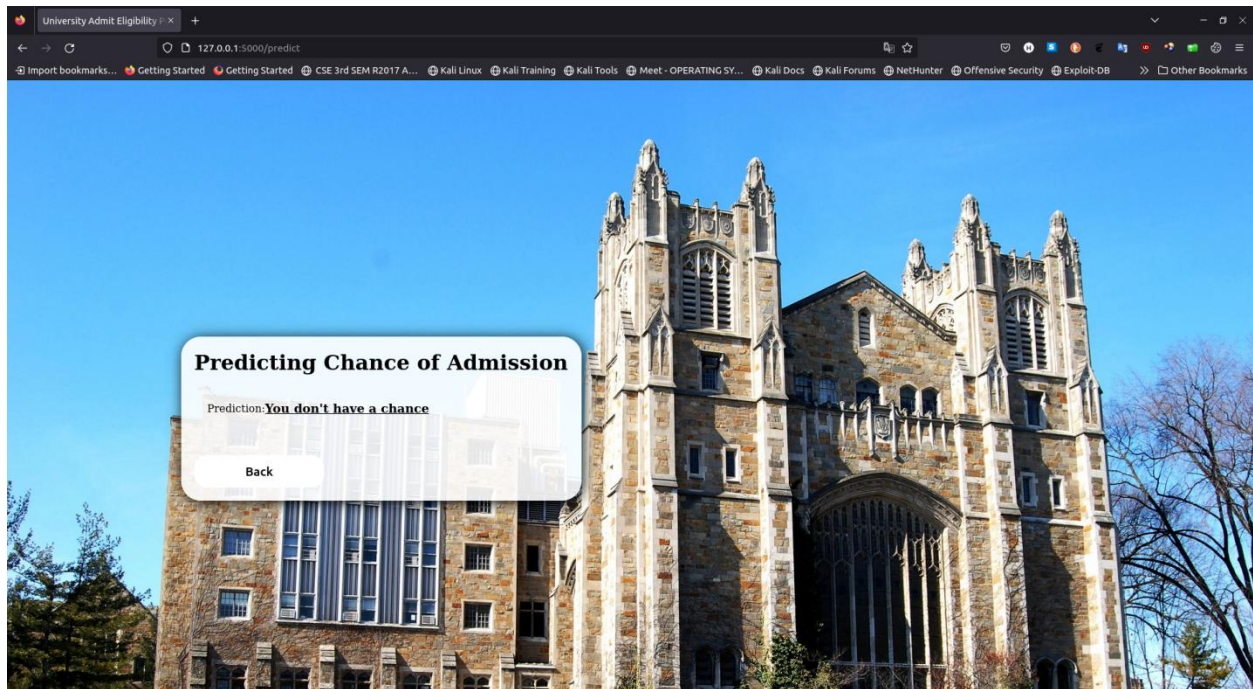
```
    app.run()
```

## OUTPUT IMAGES:









## 13.2 GITHUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-8830-1658933471>

## 13.3 DEMO LINK:

<https://drive.google.com/file/d/1ypUkgaxFIdYUEm-y5tKUFOKNWMoVABPb/view?usp=sharing>

