JEPPIAAR SRR
ENGINEERING COLLEGE

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

A PROJECT REPORT

TEAM ID: PNT2022TMID37556

Submitted by

YAARAM RAM KIRAN
AZHAGIRI.P
ABINASH.J
FRANK RAGESH.J

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING
JEPPIAAR SRR ENGINEERING COLLEGE

NOV 2022

CONTENTS

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

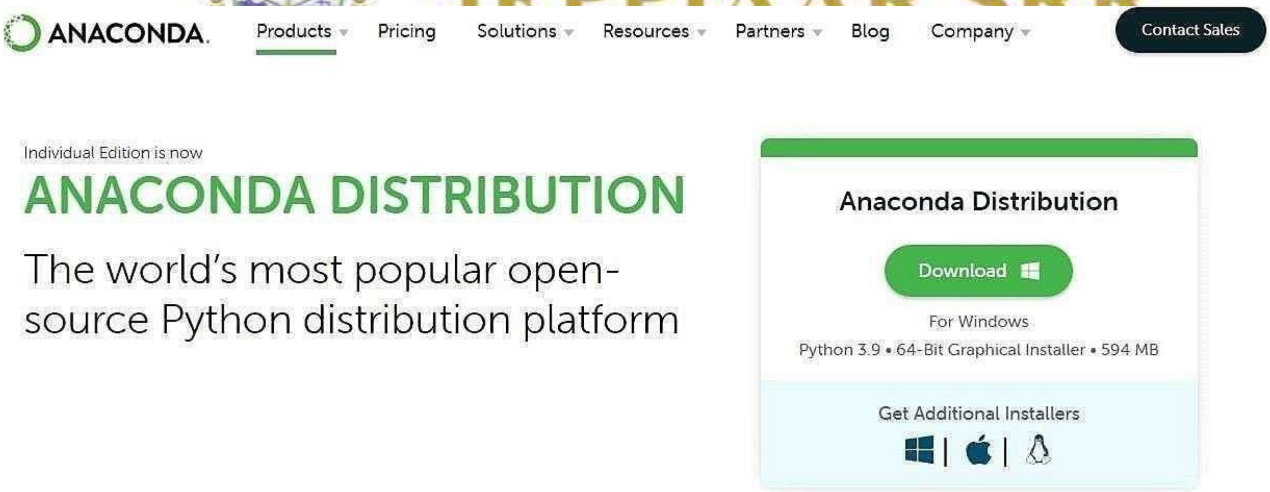
PRE-REQUISTIES

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



Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

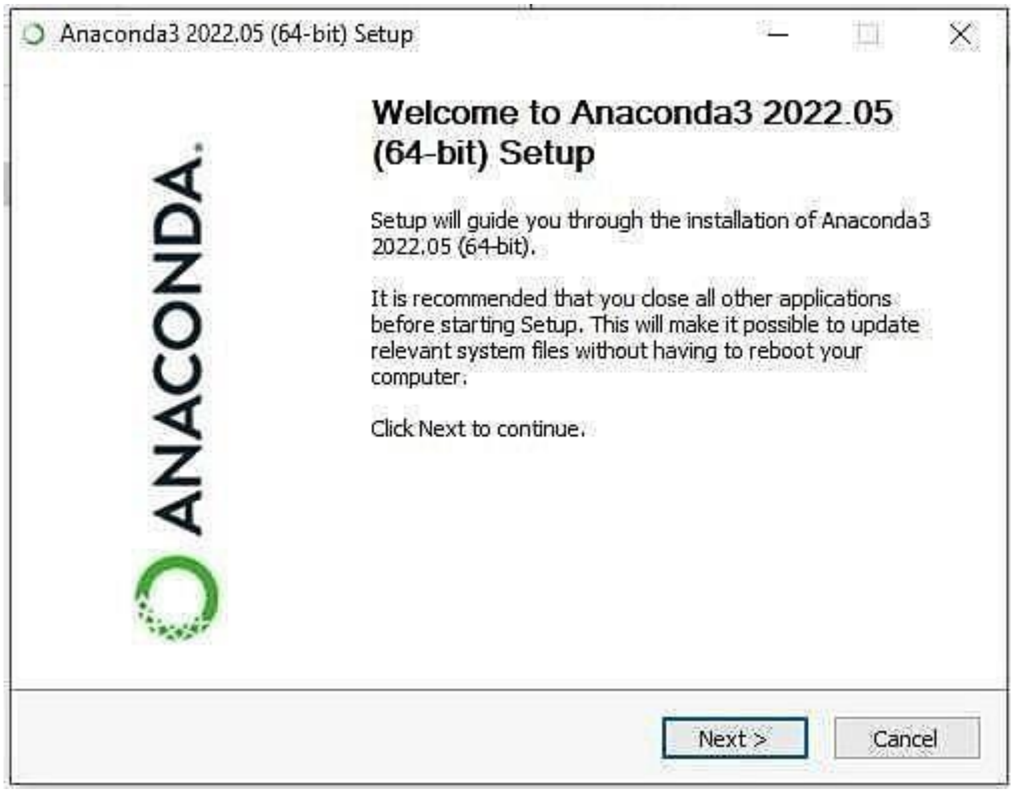
Anaconda Distribution

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

Get Additional Installers

STEP 2: Install the Anaconda



Anaconda3 2022.05 (64-bit) Setup

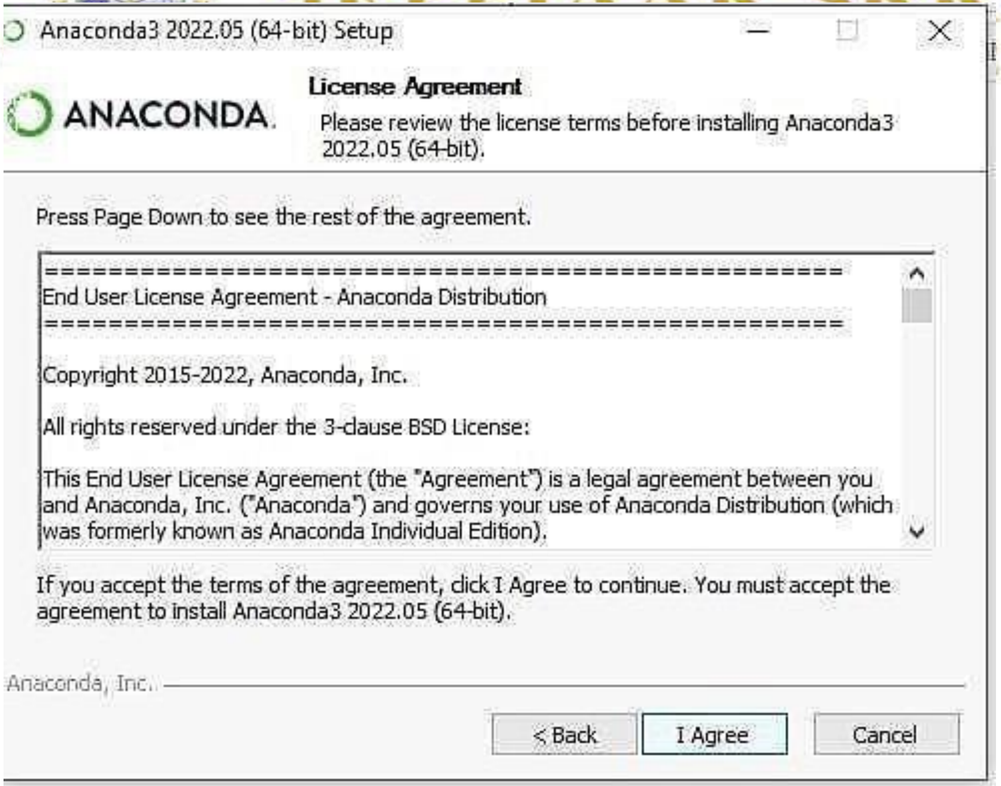
Welcome to Anaconda3 2022.05 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2022.05 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

STEP 3: Click I Agree



Anaconda3 2022.05 (64-bit) Setup

License Agreement

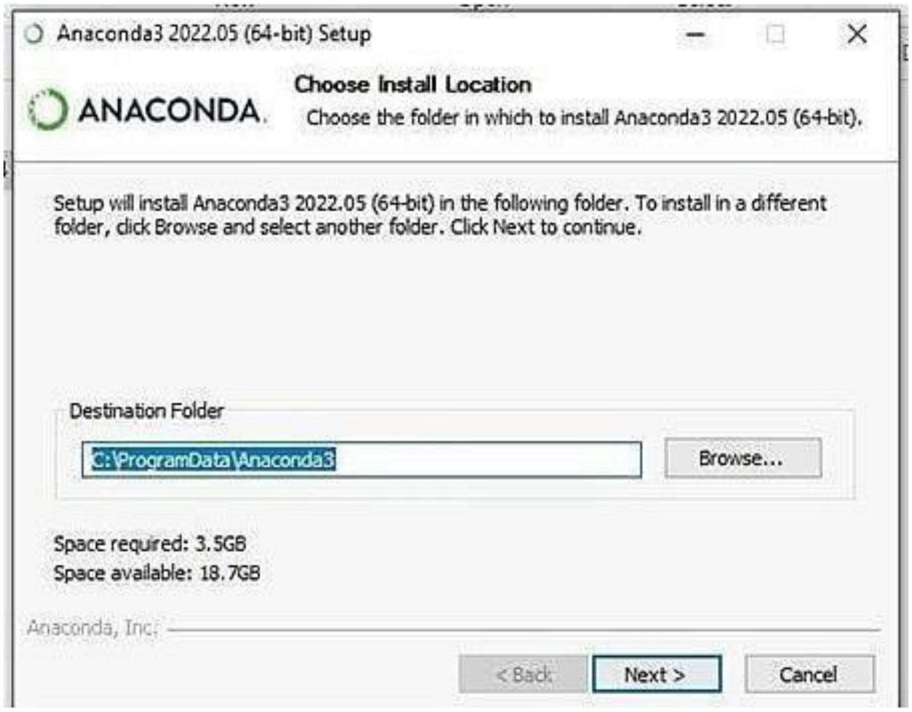
Please review the license terms before installing Anaconda3 2022.05 (64-bit).

Press Page Down to see the rest of the agreement.

```
====================================================
End User License Agreement - Anaconda Distribution
====================================================

Copyright 2015-2022, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you
and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which
was formerly known as Anaconda Individual Edition).
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2022.05 (64-bit).

STEP 4: Choose the Installation Location



Anaconda3 2022.05 (64-bit) Setup

Choose Install Location

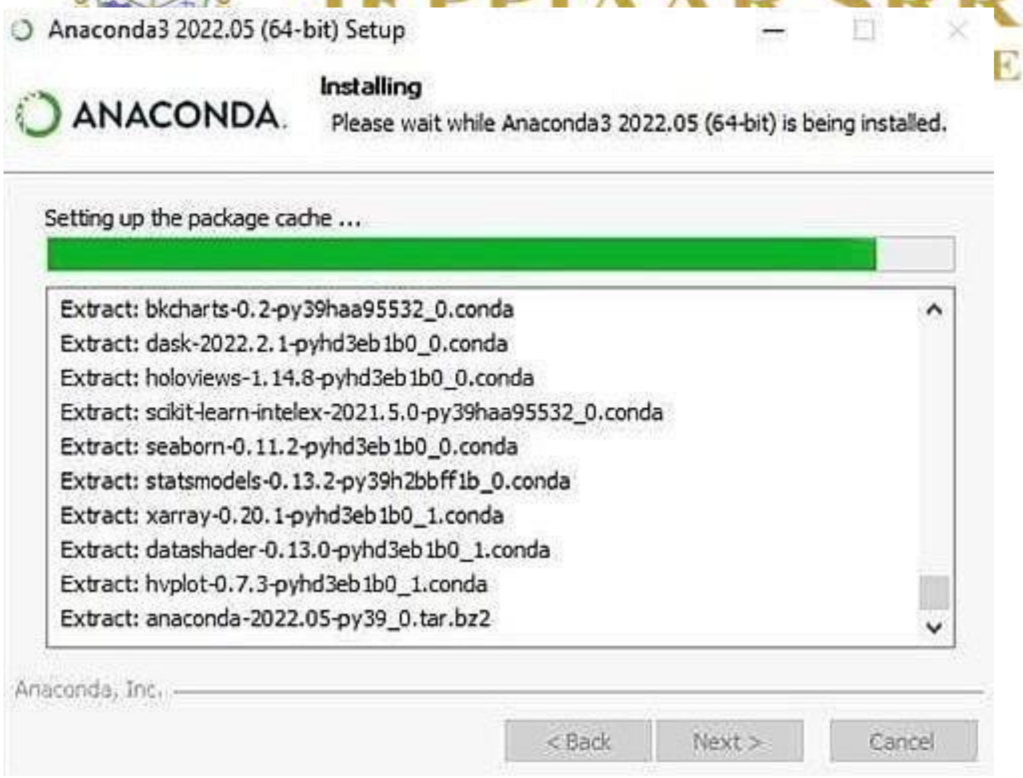
Choose the folder in which to install Anaconda3 2022.05 (64-bit).

Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\ProgramData\Anaconda3

Space required: 3.5GB
Space available: 18.7GB

STEP 5: Installing the Requiring packages



Installing

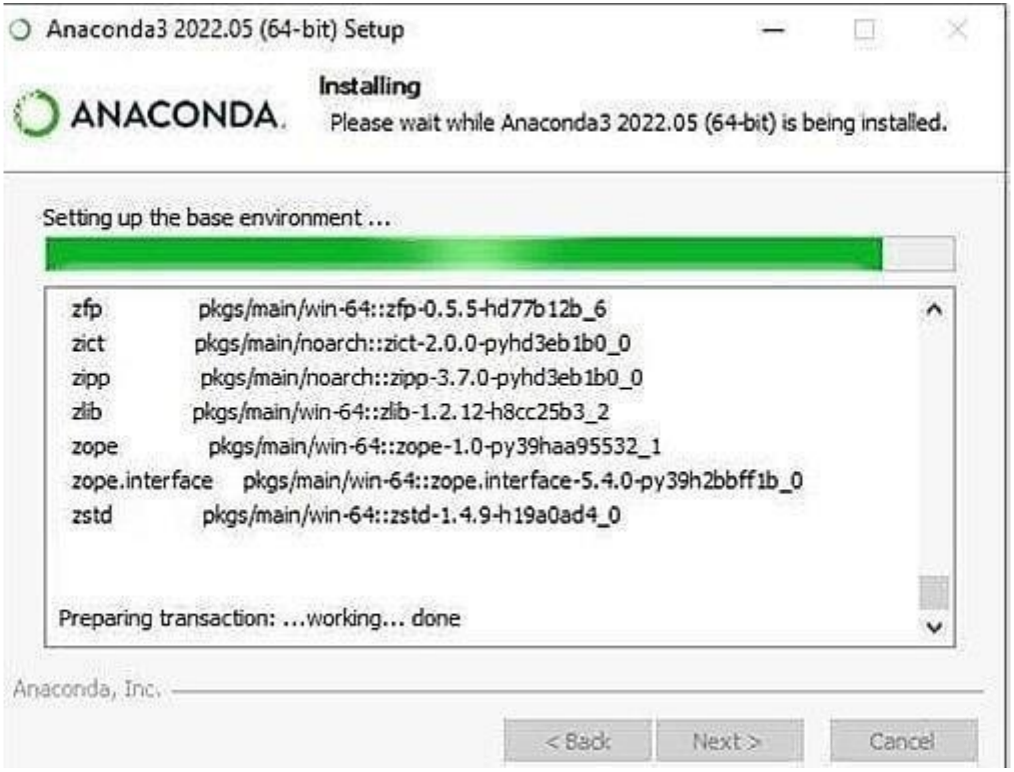
Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Setting up the package cache ...

```
Extract: bkcharts-0.2-py39haa95532_0.conda
Extract: dask-2022.2.1-pyhd3eb1b0_0.conda
Extract: holoviews-1.14.8-pyhd3eb1b0_0.conda
Extract: scikit-learn-intelex-2021.5.0-py39haa95532_0.conda
Extract: seaborn-0.11.2-pyhd3eb1b0_0.conda
Extract: statsmodels-0.13.2-py39h2bbff1b_0.conda
Extract: xarray-0.20.1-pyhd3eb1b0_1.conda
Extract: datashader-0.13.0-pyhd3eb1b0_1.conda
Extract: hvplot-0.7.3-pyhd3eb1b0_1.conda
Extract: anaconda-2022.05-py39_0.tar.bz2
```

Anaconda, Inc.

STEP 6: Setting up the base environment



Installing

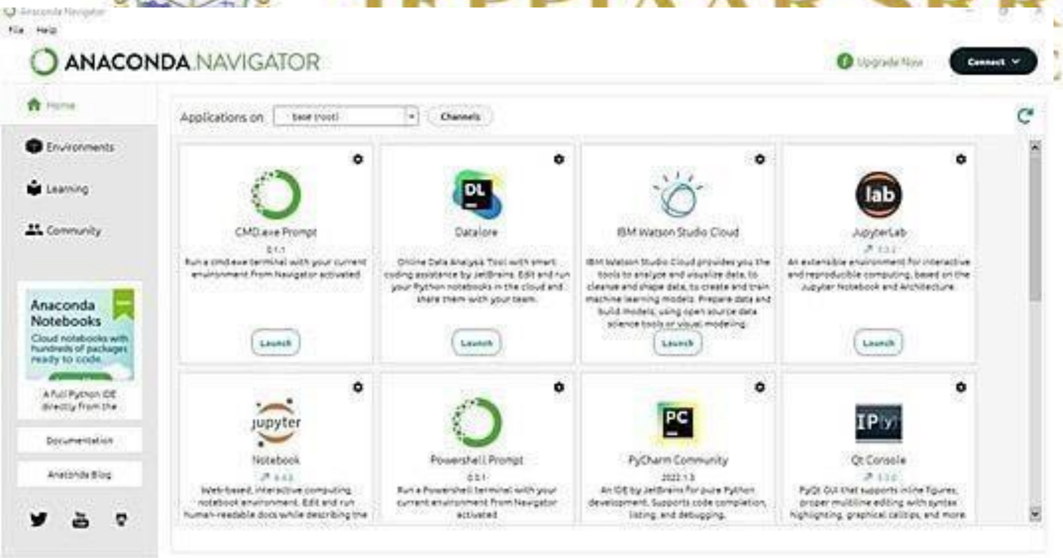
Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Setting up the base environment ...

```
zfp            pkgs/main/win-64::zfp-0.5.5-hd77b12b_6
zict           pkgs/main/noarch::zict-2.0.0-pyhd3eb1b0_0
zipp           pkgs/main/noarch::zipp-3.7.0-pyhd3eb1b0_0
zlib           pkgs/main/win-64::zlib-1.2.12-h8cc25b3_2
zope           pkgs/main/win-64::zope-1.0-py39haa95532_1
zope.interface pkgs/main/win-64::zope.interface-5.4.0-py39h2bbff1b_0
zstd           pkgs/main/win-64::zstd-1.4.9-h19a0ad4_0


Preparing transaction: ...working... done
```

Anaconda, Inc.

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



Run a cmd.exe terminal with your current environment from Navigator activated

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the

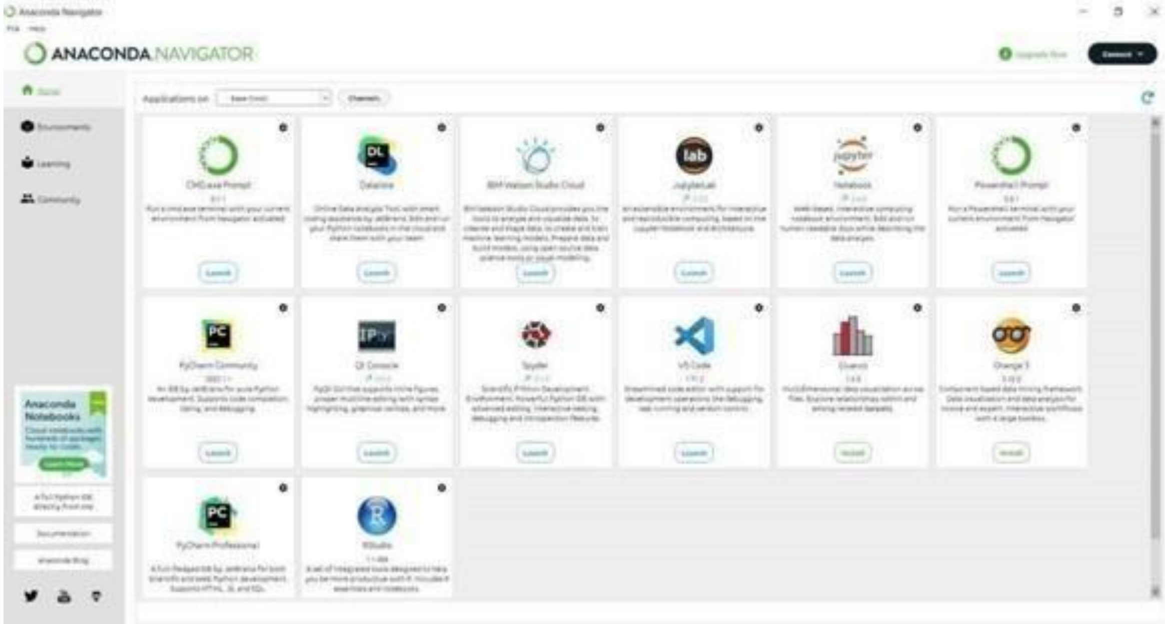
Run a Powershell terminal with your current environment from Navigator activated

An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

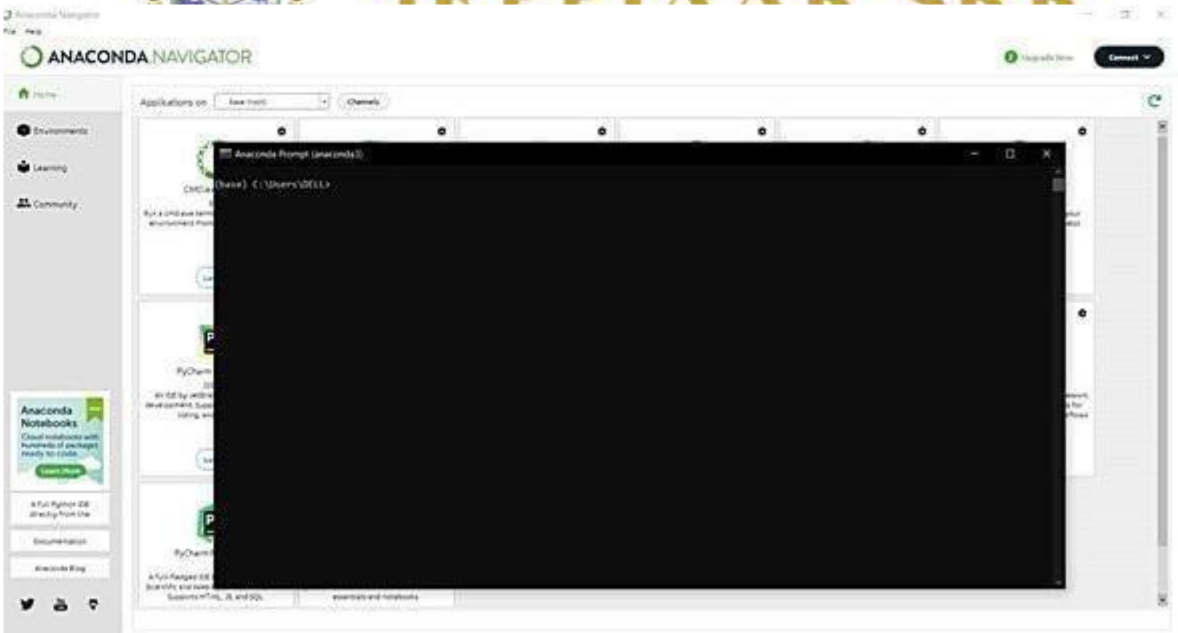
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Python packages installation:

Step 1: Open the anaconda navigator in the start menu



Step 2: Open the CMD.exe prompt



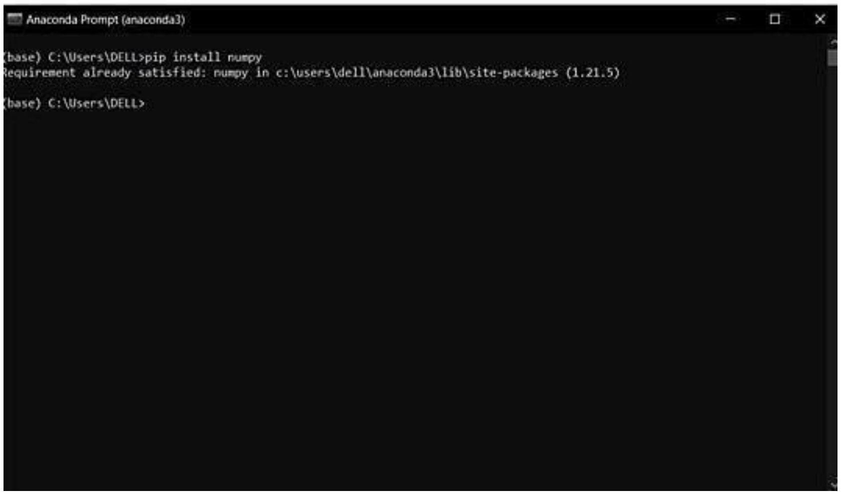
```
(base) C:\Users\DELL>
```

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.



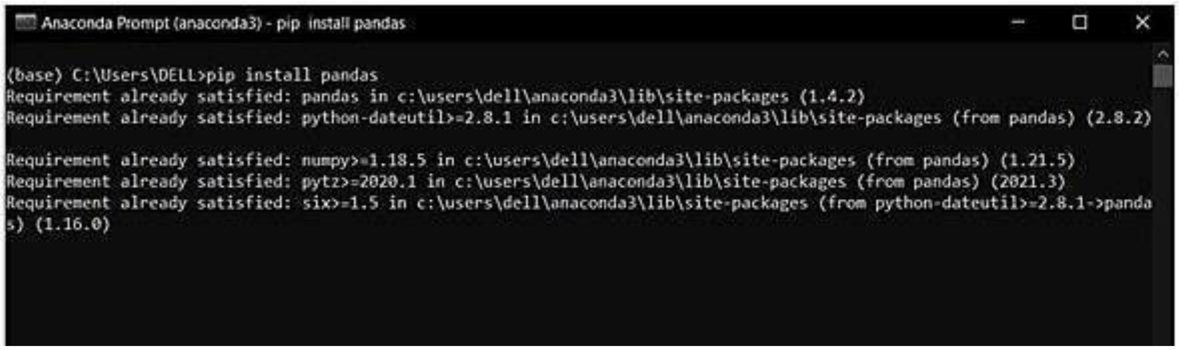
```
(base) C:\Users\DELL>pip install numpy
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (1.21.5)

(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.



```
(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->panda
s) (1.16.0)
```

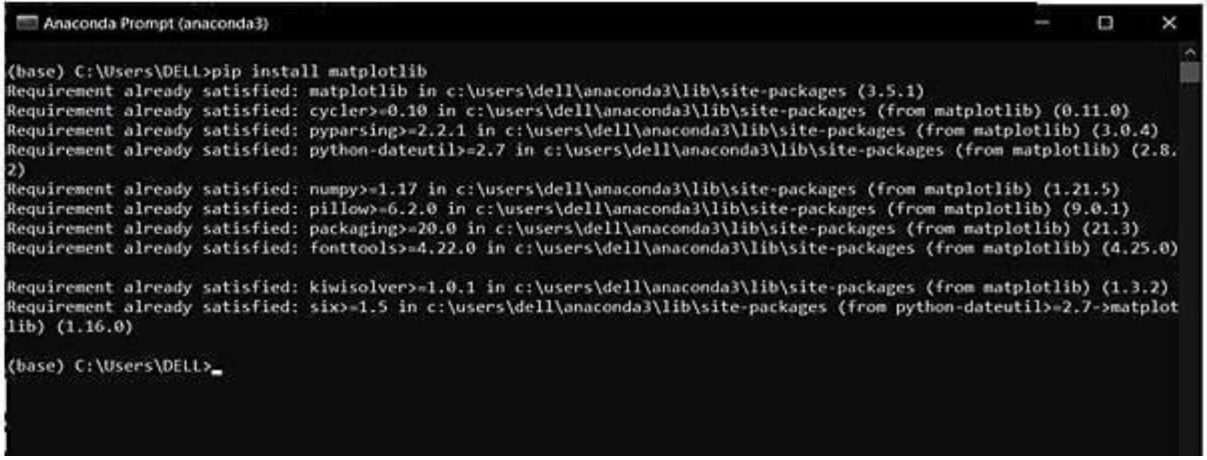
Step 5: Install the Matplotlib package.

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)

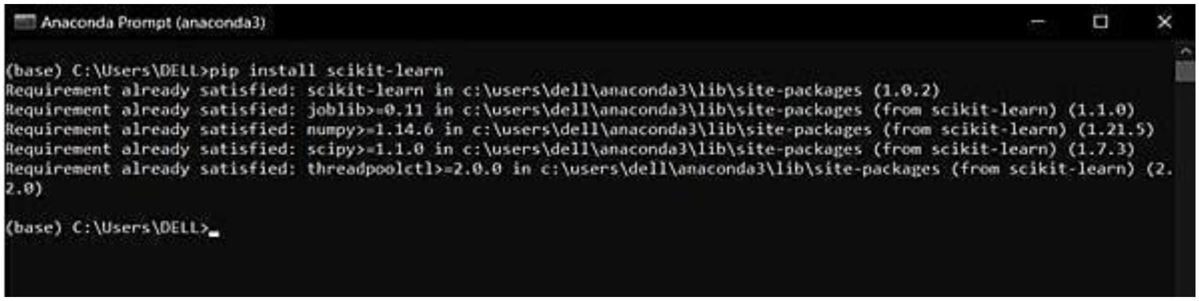
```
(base) C:\Users\DELL>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 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->matplot
lib) (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

Command: **Pip install Scikit-learn Scikit-learn:**

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.



Anaconda Prompt (anaconda3)

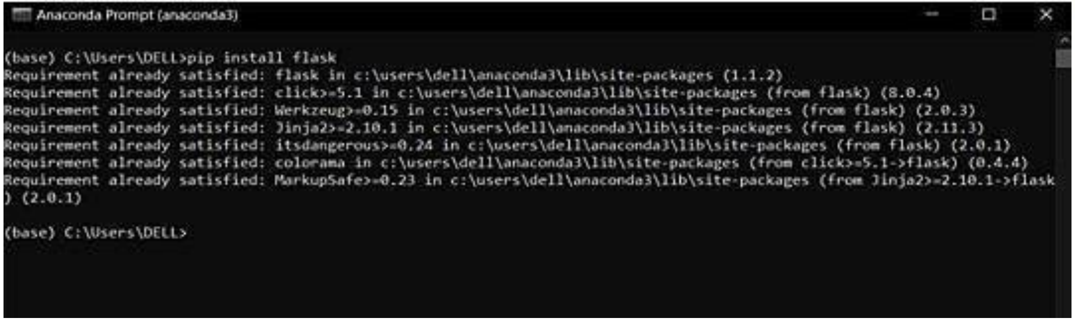
```
(base) C:\Users\DELL>pip install scikit-learn
Requirement already satisfied: scikit-learn in c:\users\dell\anaconda3\lib\site-packages (1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6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



```
(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

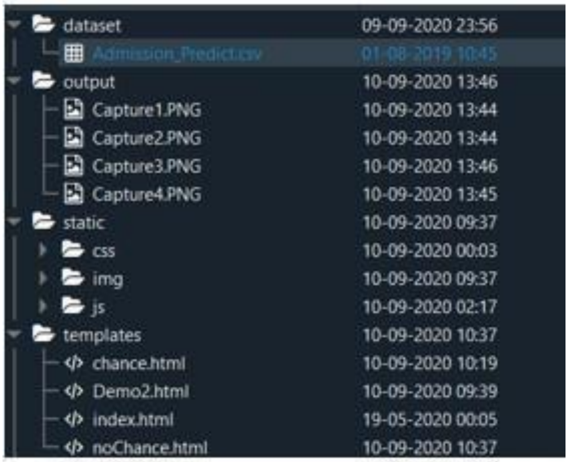
PROJECT OBJECTIVES

- 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

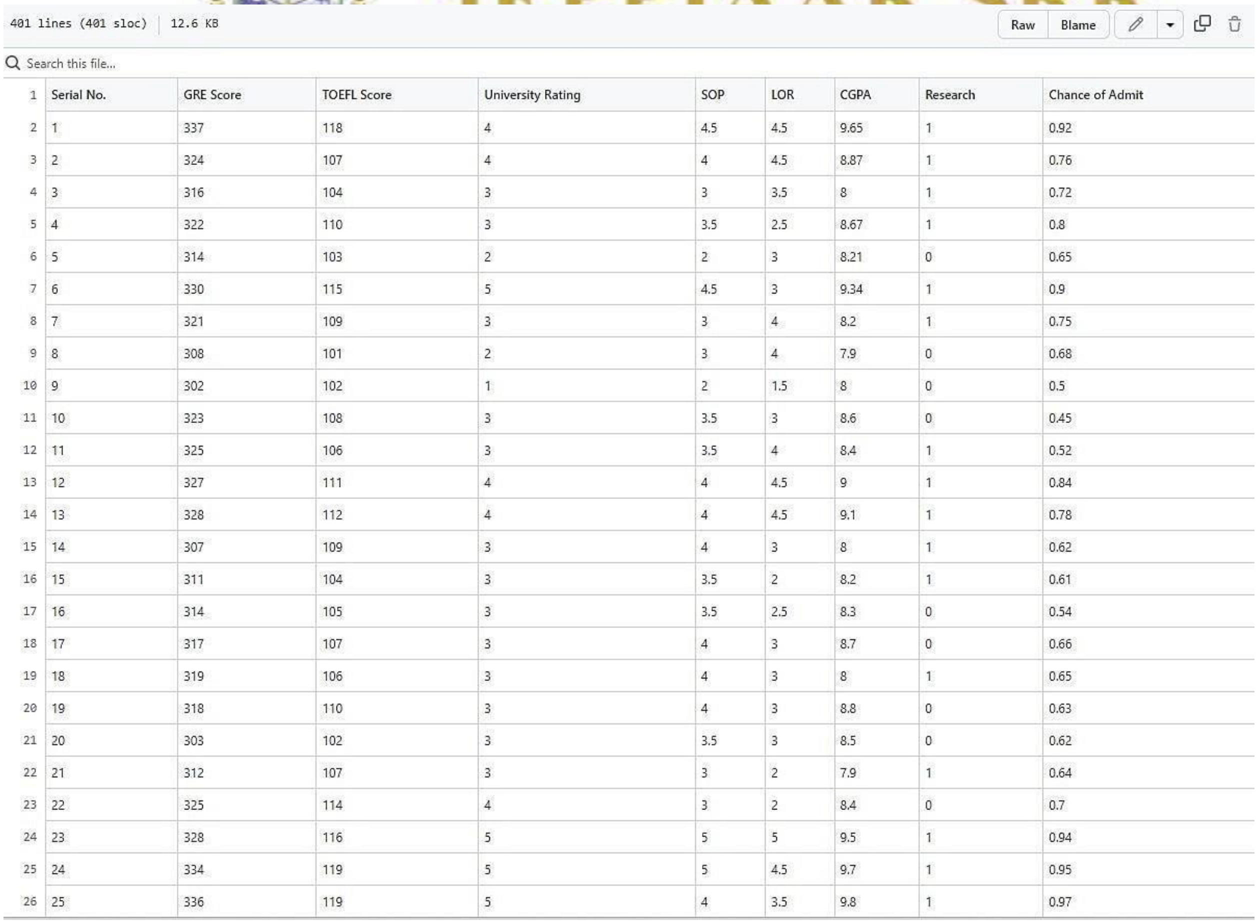
Create a Project folder that contains files as shown below



```
dataset                     09-09-2020 23:56
  Admission_Predict.csv     01-08-2019 10:45
output                      10-09-2020 13:46
  Capture1.PNG              10-09-2020 13:44
  Capture2.PNG              10-09-2020 13:44
  Capture3.PNG              10-09-2020 13:46
  Capture4.PNG              10-09-2020 13:45
static                      10-09-2020 09:37
  css                       10-09-2020 00:03
  img                       10-09-2020 09:37
  js                        10-09-2020 02:17
templates                   10-09-2020 10:37
  chance.html               10-09-2020 10:19
  Demo2.html                10-09-2020 09:39
  index.html                19-05-2020 00:05
  noChance.html             10-09-2020 10:37
```

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 information for machine learning models. Kaggle is probably the most visited website used for information gathering. Collect the dataset or Create the dataset.



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

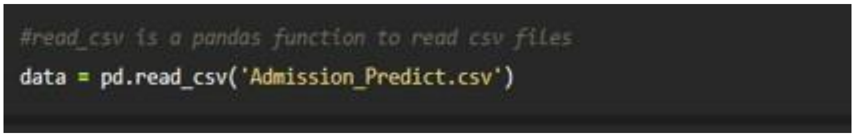
Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. ● **Matplotlib-**

Visualisation with python. It is a comprehensive library for creating static, animated, and interactive visualizations in Python

Reading the Datasets:

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

Let’s load a .csv data file into pandas usingread_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

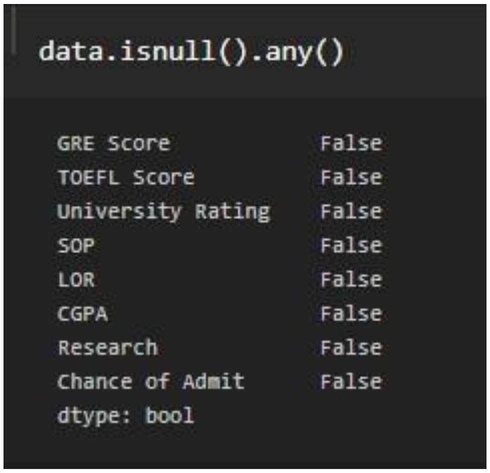
1.Serial No.
2. GRE Score
3.TOEFL Score
4.University Rating
5.SOP
6.LOR
7.CGPA
8.Chance of Admit

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.

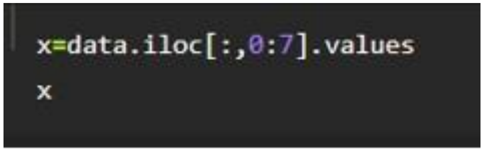
To visualize the dataset we need libraries called Matplotlib and Seaborn. The Matplotlib library is a Python 2D plotting library that allows you to generate plots, scatter plots, histograms, bar charts etc.

Splitting Dependent And Independent Columns:

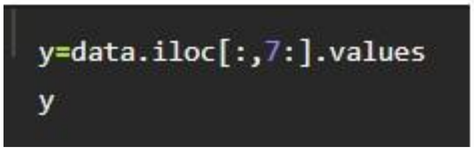
We need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

- To read the columns, we will useiloc 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.



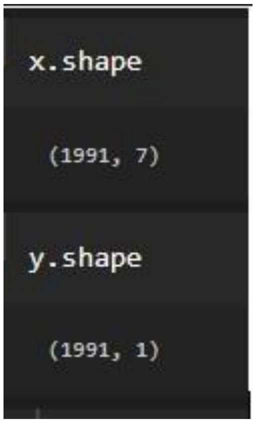
```
x=data.iloc[:,0:7].values
x
```



```
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 thefully 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.

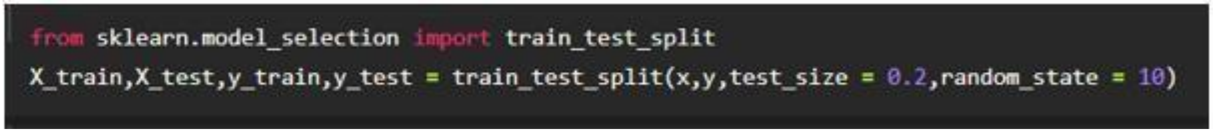
To help us with this task, the Scikit library provides a tool, called the Model Selection library.

There is a class in the library which is, ‘train_test_split.’ Using this we can easily split the dataset into the training and the testing datasets in various proportions.

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

- x_train
- x_test
- y_train
- 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

➢ https://ieeexplore.ieee.org/document/9418279

Abstract:
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.
References: MS Acharya, A Armaan and AS Antony, "A comparison of regression models for prediction of graduate admissions", 2019.

➢ https://ieeexplore.ieee.org/document/9410717

Abstract:
Students applying for admissions to universities find it difficult to understand whether they have good chances of getting admission in a university or not. Keeping this in focus, we have used logistic regression techniques that have gained attention in software engineering field for its ability to be used for predictions. This is a novel work on a university admissions predictor using which students can evaluate their competitiveness for getting admission at a university.
References: M. Fatima and M. Pasha, "Survey of machine learning algorithms for disease diagnostic", *Journal of Intelligent Learning Systems and Applications*, vol. 9, no. 01, pp. 1, 2017.

- ➢ https://ieeexplore.ieee.org/document/6416521

Abstract:

This paper presents a new college admission system using hybrid recommender based on data mining techniques and knowledge discovery rules, for tackling college admissions prediction problems. This is due to the huge numbers of students required to attend university colleges every year. The proposed HRSPCA system consists of two cascaded hybrid recommenders working together with the help of college predictor, for achieving high performance.

References: G. Ganapathy, and K. Arunesh, "Models for Recommender Systems in Web Usage Mining Based on User Ratings" Proceedings of the World Congress on Engineering, Vol. I WCE 2011.

- ➢ https://dl.acm.org/doi/10.1145/3388818.3393716

Abstract:

With the increase in the number of graduates who wish to pursue their education, it becomes more challenging to get admission to the students' dream university. Newly graduate students usually are not knowledgeable of the requirements and the procedures of the postgraduate admission and might spent a considerable amount of money to get advice from consultancy organizations to help them identify their admission chances.

References: E. Roberts, "using machine learning and predictive modeling to assess admission policies and standards," 2013.

- ➢ https://medium.com/@jigar18011999/university-predictor-by-machine-learning-2d880e9f3a3

Abstract:

This article describes the architecture and algorithms of the proposed system. ANN, decision trees, and logistic regression were used to find admissions for a particular student. ML models take into account various parameters such as GRE and TOEFL scores, SOP, and LOR. Finally, after evaluation, the authors state that decision trees are the most accurate among the tree algorithms used.

- https://github.com/satwik2663/Machine-Learning-Graduate-Studuent-Admission-Predictor

Abstract:

Today, there are many students who travel to USA to pursue higher education. It is necessary for the students to know what are their chances of getting an admit in the universities. Also, universities manually check and count the total number of applicants who could get an admit into university. These methods are slow and certainly not very consistent for students and universities to get an actual result. This method is also prone to human error and thus accounts for some inaccuracies. Since the frequency of students studying abroad has increased, there is a need to employ more efficient systems which handle the admission process accurately from both perspectives.

- https://github.com/anjanatiha/University-Admission-Match-Predictor

Abstract:

1. Analyzed university admission statistics.
2. Developed tools for matching university (in percentile) using CGPA, GRE (Verbal, Quantitative, Analytical Writing) scores.

- https://github.com/karanwadhwa/dd-admission-predictor

Abstract:

This system was originally developed only for Engineering College Admissions in Maharashtra,India but can essentially be adapted for other streams too. The purpose of it is to build a system to predict the users chances for getting into a certain college.

2.3 Problem Statement Definition

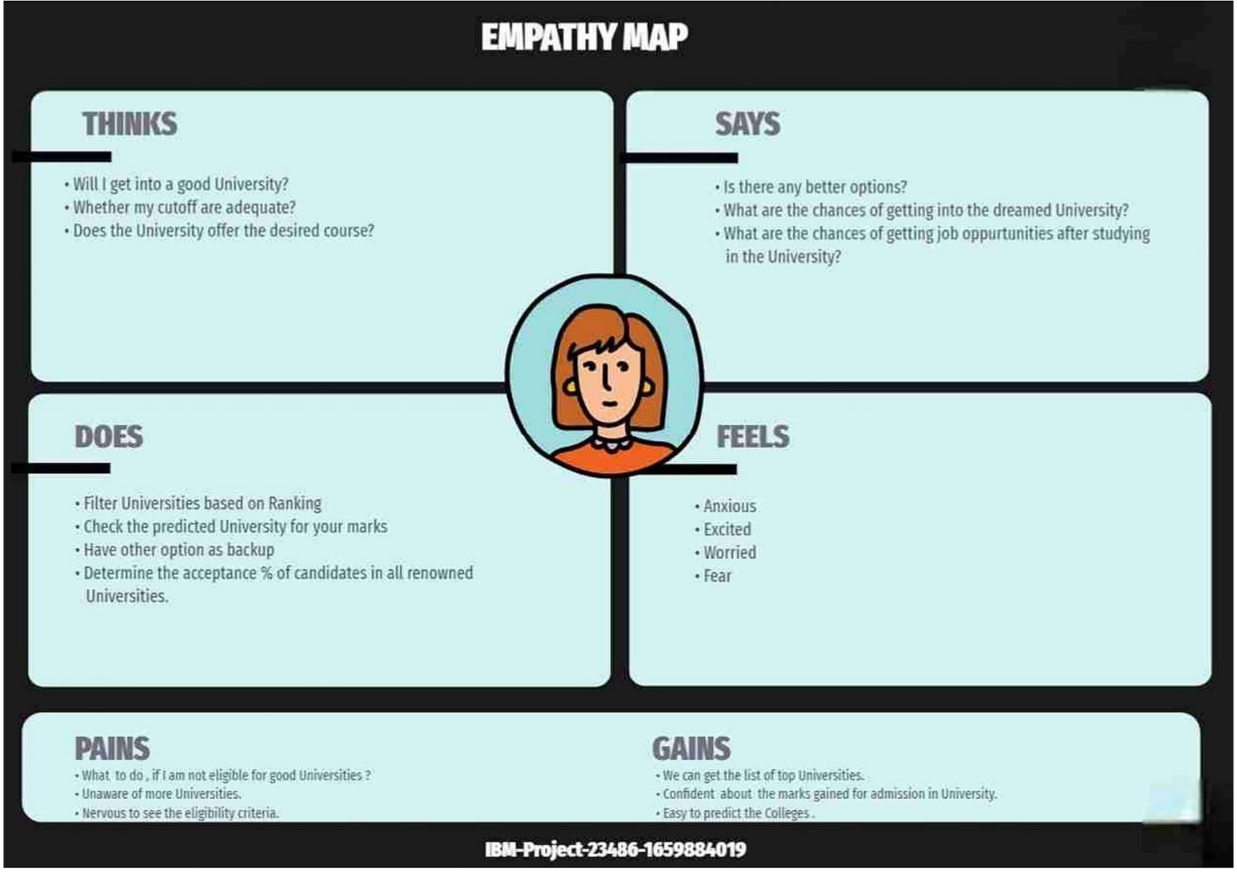
Problem Statement(PS)	I am (Customer)	I'm trying to	But	Because	Which makes mefeel
PS-1	Student	I am looking university for long time	Still I am facing difficulties to select the university that is having good environment	University is in Chennai but I want in my Home town	Difficult to find the university.
PS-2	Student	I am Searching Universityon my phone	It is difficultto find and it takes long time	It is difficult tosearch in offline	Exhausted

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



EMPATHY MAP

THINKS

- Will I get into a good University?
- Whether my cutoff are adequate?
- Does the University offer the desired course?

SAYS

- Is there any better options?
- What are the chances of getting into the dreamed University?
- What are the chances of getting job oppurtunities after studying in the University?

DOES

- Filter Universities based on Ranking
- Check the predicted University for your marks
- Have other option as backup
- Determine the acceptance % of candidates in all renowned Universities.

FEELS

- Anxious
- Excited
- Worried
- Fear

PAINS

- What to do , if I am not eligible for good Universities ?
- Unaware of more Universities.
- Nervous to see the eligibility criteria.

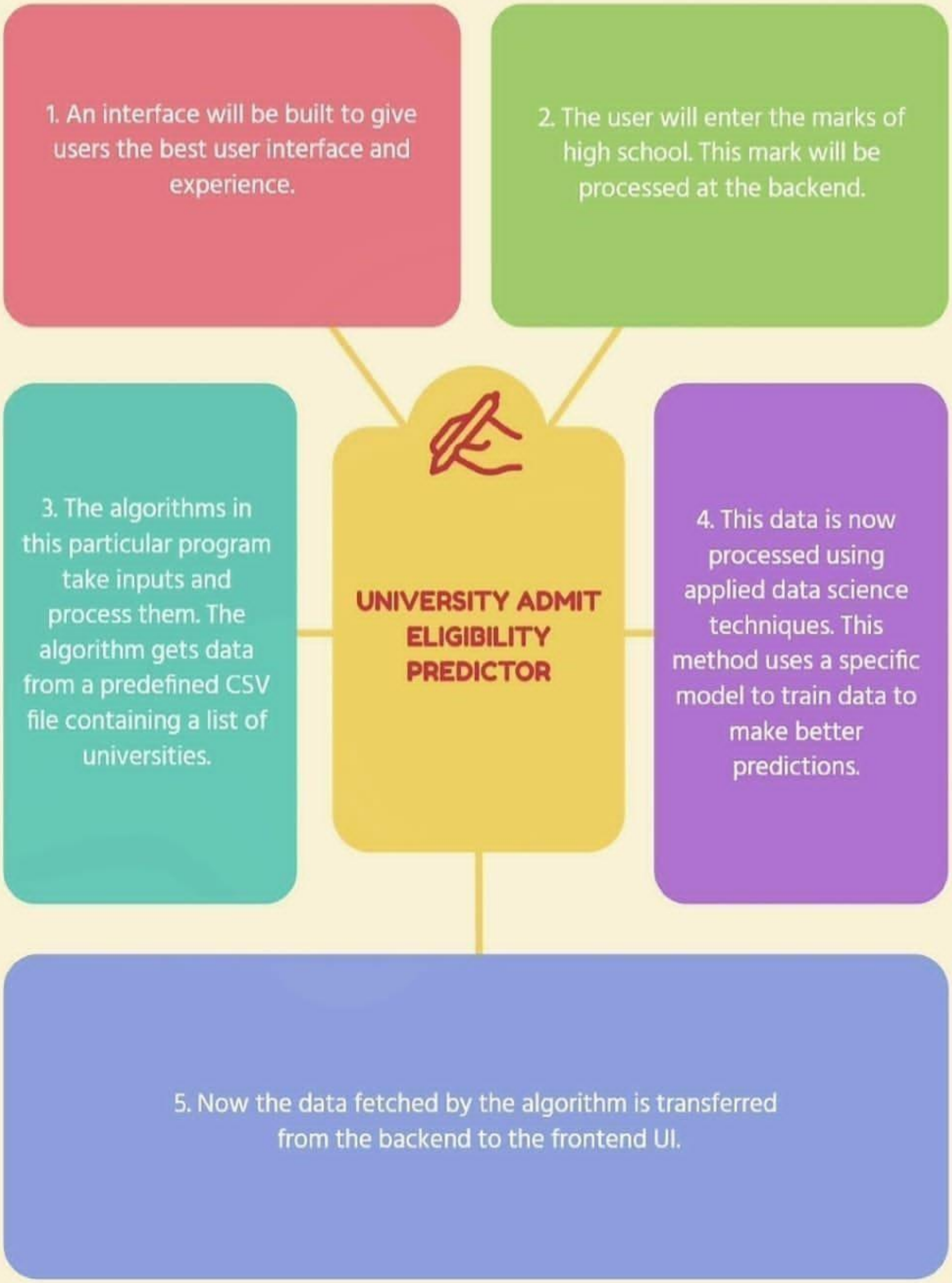
GAINS

- We can get the list of top Universities.
- Confident about the marks gained for admission in University.
- Easy to predict the Colleges .

IBM-Project-23486-1659884019

3.2 Ideation & Brainstorming

BRAINSTORMING



1. An interface will be built to give users the best user interface and experience.

2. The user will enter the marks of high school. This mark will be processed at the backend.

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

3. The algorithms in this particular program take inputs and process them. The algorithm gets data from a predefined CSV file containing a list of universities.

4. This data is now processed using applied data science techniques. This method uses a specific model to train data to make better predictions.

5. Now the data fetched by the algorithm is transferred from the backend to the frontend UI.

IBM-Project-23486-1659884019

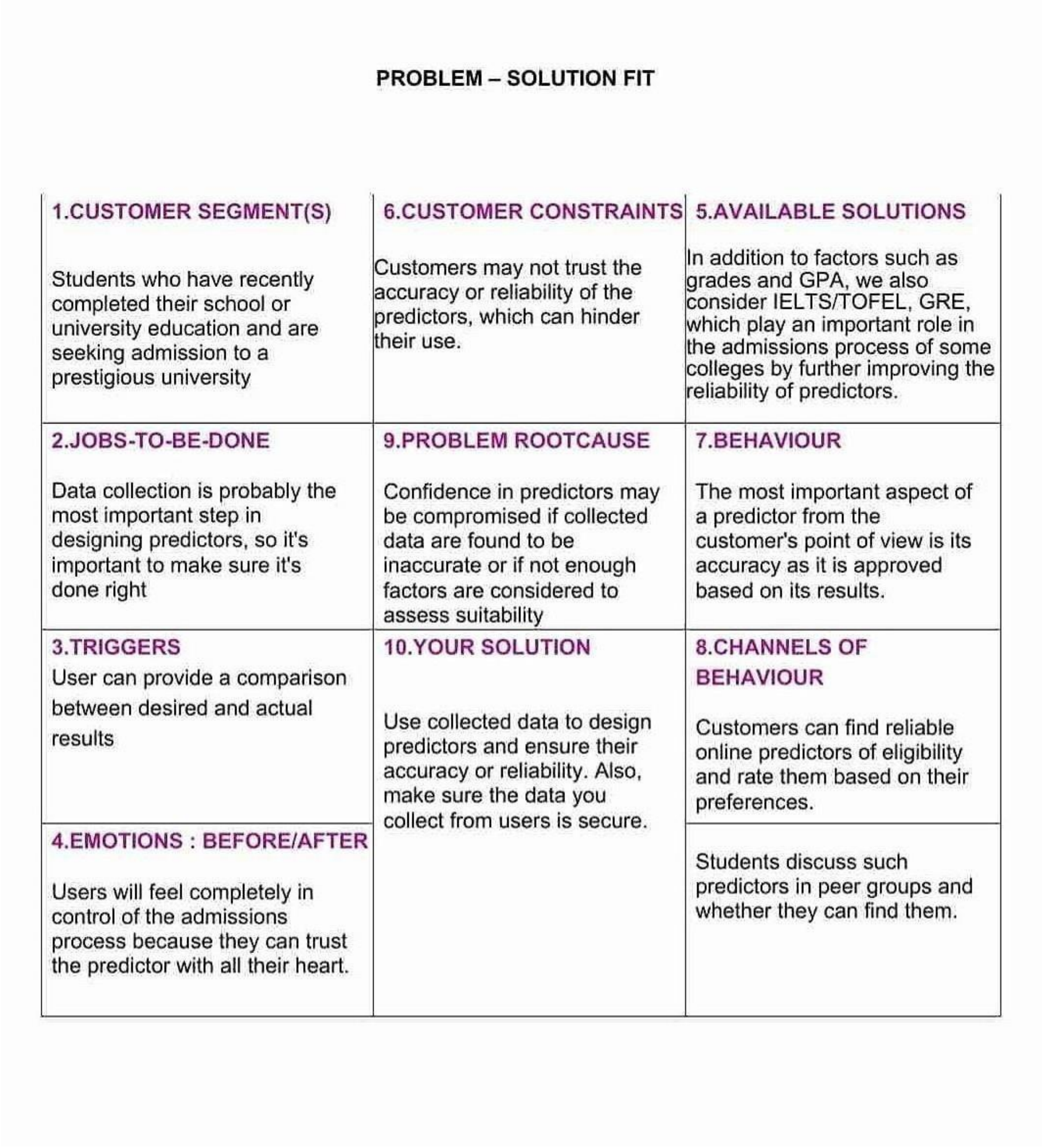
3.3 Proposed Solution

1.	Problem Statement (Problem to besolved)	Students often worry about their chances of getting into college. The goal of this project is to help profile shortlisted college students. Predicted results give them a good ideaof their likelihood of getting into a particular college. This analysis is also useful for students preparing or planning to prepare for a betterimage must. It also aimsto connect students and universities directly, withoutintermediaries.
2.	Idea/ Solution description	This project aims to calculate the likelihood of admission to aparticular graduate schoolafter evaluating a candidate's profile. The main attributes considered in decision makingare: 1. GRE &TOEFL Scores 2. Undergraduate CGPA 3. SOP &LOR 4. Corporate WorkExperience/Research ExperienceExtracurricular Activities 5. Extra-curriculars Determine Acceptance Rate, Logistic Regression, Multi linear, Usea variety of ML models such as regression, decision trees, and random forests, and use performance metrics such as

		accuracy score, precision, and retrieval to evaluate whichmodel has the best accuracy.
3.	Novelty / Uniqueness	● We plan to develop a new hybrid model based on deep learning that is more accurate than existing traditional MLmodels. ● Students oftenhave trouble narrowing down whichcolleges to apply to.
4.	Social / Impact Customer Satisfaction	● Students oftenstruggle to narrowdown which colleges to apply to and wonderif their profile matches the requirements of a particular college. ● In addition, the cost of applying to universities is very high,and it is important for students to narrow down their universities based on their profile. ● University Admit Eligibility Predictor Systems are very useful in determining the likelihood thata

		student willbeadmitted to a particular college. ● The system reduces reliance on expensive educational consultancies to analyze candidate profiles and determine which colleges to apply to.
5.	Business Model (Revenue Model)	● Advertisements of different universities could be placedinthe web-app to generate revenue through ads. ● In the future, a separate premium plan could be created where the students candirectly interact withthe professors and alumniof the university through video calls.
6.	Scalability of theSolution	● Future updates will allow candidates, faculty, students, andalumni to interact and have a chat area where candidates can get their questions answered quickly. ● Cloud-based storage (IBM Cloud, AWS, GCP, AZURE) and NoSQLdatabases (MongoDB, Redis,etc.) to be abletohandle large amounts of data (both applicant and university data) in the future.) can usetraditional RDBMS storage ● Alternatively, if the numberof users usingyour website grows exponentially over time, you can consider distributedbig dataprocessing techniques.

3.4 Problem Solution Fit



PROBLEM – SOLUTION FIT

1.CUSTOMER SEGMENT(S) Students who have recently completed their school or university education and are seeking admission to a prestigious university	**6.CUSTOMER CONSTRAINTS** Customers may not trust the accuracy or reliability of the predictors, which can hinder their use.	**5.AVAILABLE SOLUTIONS** In addition to factors such as grades and GPA, we also consider IELTS/TOFEL, GRE, which play an important role in the admissions process of some colleges by further improving the reliability of predictors.
2.JOBS-TO-BE-DONE Data collection is probably the most important step in designing predictors, so it's important to make sure it's done right	**9.PROBLEM ROOTCAUSE** Confidence in predictors may be compromised if collected data are found to be inaccurate or if not enough factors are considered to assess suitability	**7.BEHAVIOUR** The most important aspect of a predictor from the customer's point of view is its accuracy as it is approved based on its results.
3.TRIGGERS User can provide a comparison between desired and actual results	**10.YOUR SOLUTION** Use collected data to design predictors and ensure their accuracy or reliability. Also, make sure the data you collect from users is secure.	**8.CHANNELS OF BEHAVIOUR** Customers can find reliable online predictors of eligibility and rate them based on their preferences.
4.EMOTIONS : BEFORE/AFTER Users will feel completely in control of the admissions process because they can trust the predictor with all their heart.		Students discuss such predictors in peer groups and whether they can find them.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through username and passwordLogin through Gmail Login through LinkedIN
FR-4	Administration work	Check qualified candidate detail Make allotment
FR-5	Admission Details	Check seat availability Check college infrastructure Check fees details
FR-6	Local counsellor	Issue the final allotment order

4.2 Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description

NFR-1	**Usability**	a. A logical interface is essential to make easy use ofsystem, speeding up common tasks. b. The product could be used by two categories of people mainly administrator category and otherusers.
NFR-2	**Security**	Some of the factors that are identified to protectthe software from accidental or malicious access,use, modification, destruction, or disclosure are described below: a. Keep specific log or history data sets. b. Utilize certain cryptographic techniques. c. Restrict the no of systems that can access the online admission system site. This could be done only by registering the systems physical addresses

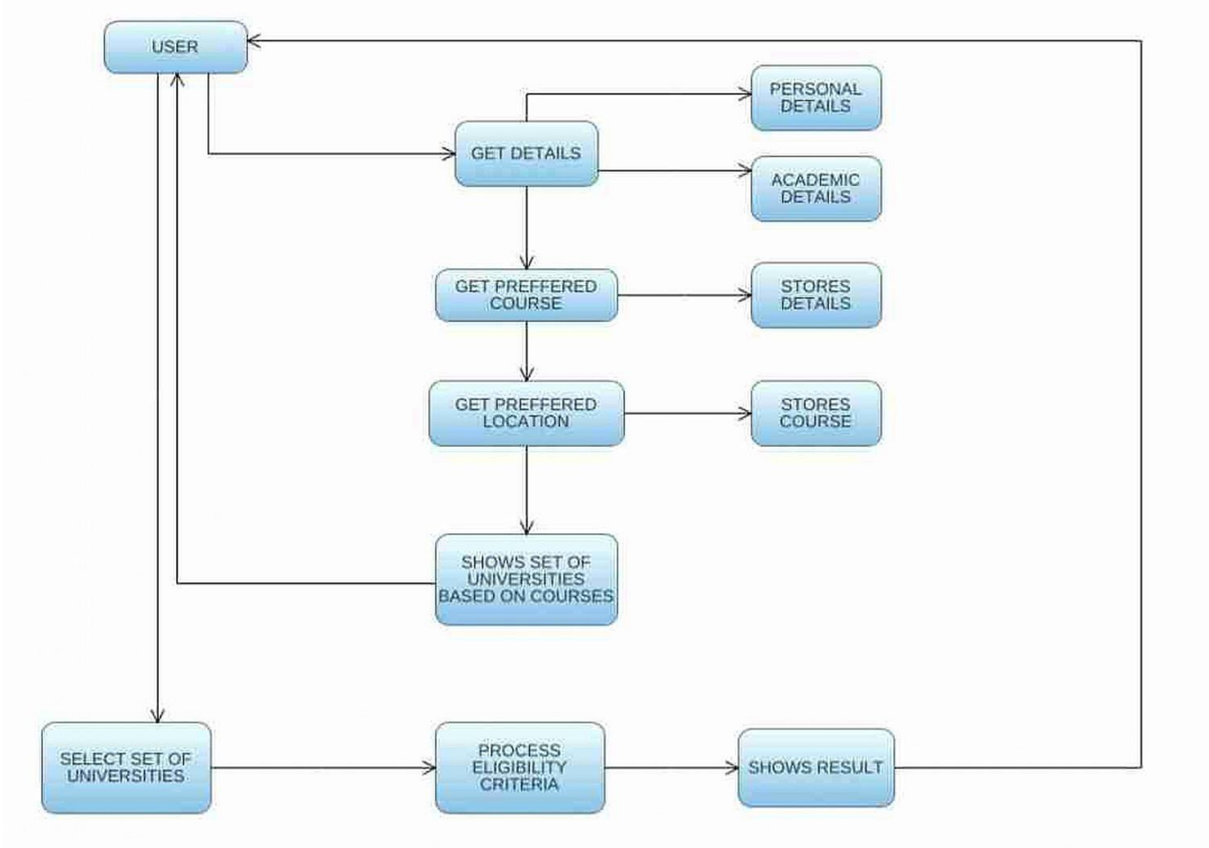
		before using them for online admission process. a. Check data integrity for critical variables. b. Every user should be licensed to use the systemunder any of the four categories provided i.e. either verifier or advisor or local counsellor or administrator. c. Communication needs to be restricted when the application is validating the user or license.
NFR-3	**Reliability**	a. All data storage for user variables will be committed to the database at the time of entry. b. Data corruption is prevented by applying thepossible backup procedures and techniques.
NFR-4	**Performance**	a. The database should be able to accommodate aminimum of 10,000 records of students. b. At any instant the system should support use ofmultiple users at a time. c. Availability results of the requested college should be presented to the student in max of twoseconds, so retrieving of data should be reliable.

		d. As each student will be given a maximum timeof 10min, accessing from the database
		should be done at relevant speed.
NFR-5	**Availability**	The system should available at all the time meaning that the user can access easily. Increase ofthe hardware and data base failure a replacement page will be show and for database back should be retrieved from data folder.

NFR-6	**Scalability**	Assesses the highest workloads under which the system will still meet the performance Deals withthe measure of the system's response time under different load conditions requirements. Example: The system must be scalable enough to support 1,000,000 visits at the same time while maintaining optimal performance.

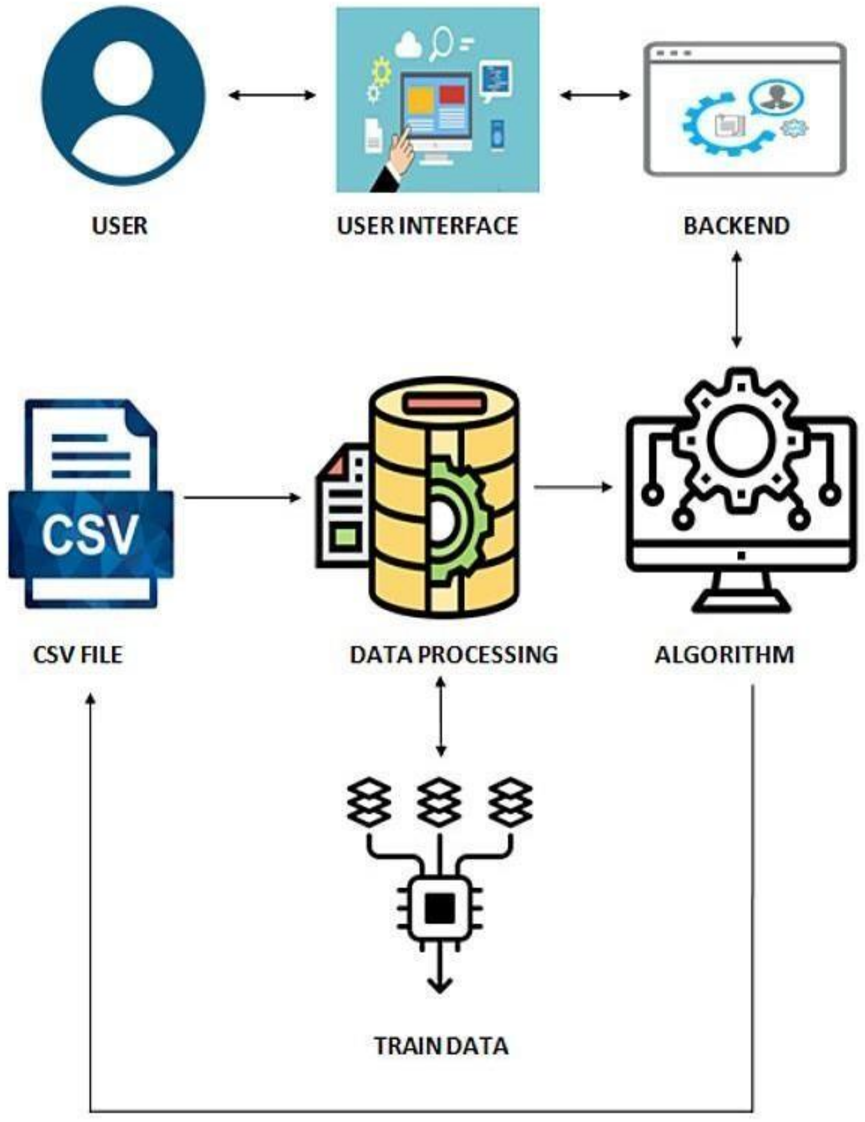
5. PROJECT DESIGN

5.1 Data Flow Diagram



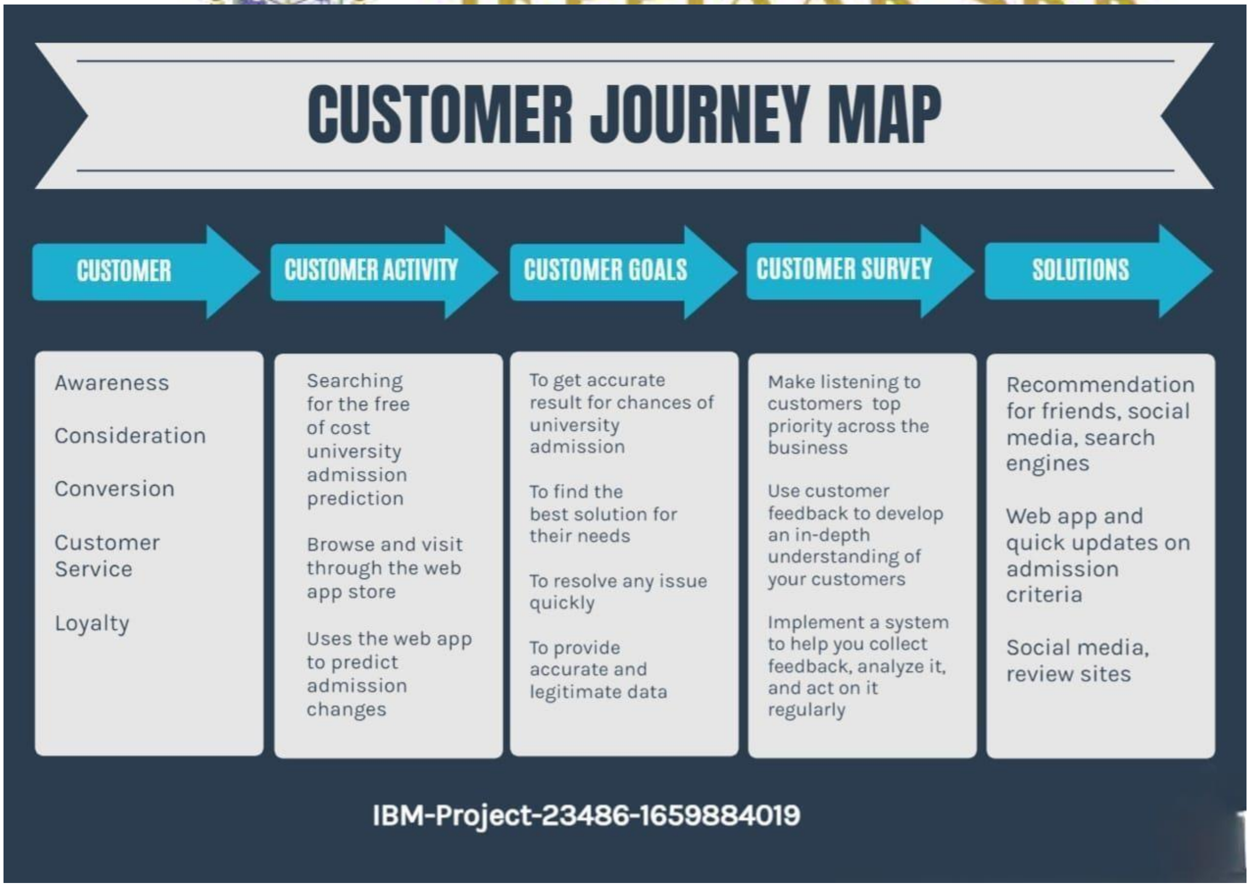
5.2 Solution & Technical Architecture

WORKING ALGORITHM



IBM-Project-23486-1659884019

5.3 User Stories



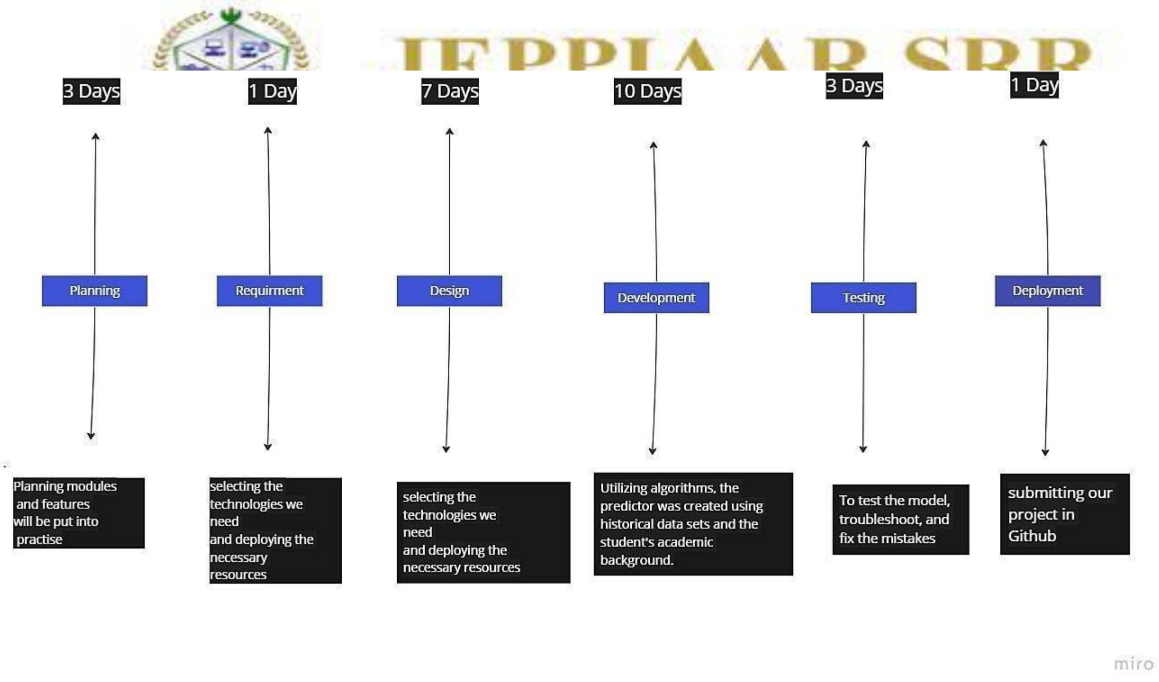
CUSTOMER JOURNEY MAP

CUSTOMER	CUSTOMER ACTIVITY	CUSTOMER GOALS	CUSTOMER SURVEY	SOLUTIONS
Awareness Consideration Conversion Customer Service Loyalty	Searching for the free of cost university admission prediction Browse and visit through the web app store Uses the web app to predict admission changes	To get accurate result for chances of university admission To find the best solution for their needs To resolve any issue quickly To provide accurate and legitimate data	Make listening to customers top priority across the business Use customer feedback to develop an in-depth understanding of your customers Implement a system to help you collect feedback, analyze it, and act on it regularly	Recommendation for friends, social media, search engines Web app and quick updates on admission criteria Social media, review sites

IBM-Project-23486-1659884019

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation


3 Days
1 Day
7 Days
10 Days
3 Days
1 Day
Planning
Requirment
Design
Development
Testing
Deployment
Planning modules and features will be put into practise
selecting the technologies we need and deploying the necessary resources
selecting the technologies we need and deploying the necessary resources
Utilizing algorithms, the predictor was created using historical data sets and the student's academic background.
To test the model, troubleshoot, and fix the mistakes
submitting our project in Github
miro

6.2 Sprint Delivery Schedule

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story/ Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Registration	USN-1	As a user, you can register in the application by entering your emailaddress, password,	2	High	Jayapradha B

			and confirming thepassword			
Sprint-1		USN-2	As a user, you will receive a confirmation emailafter registering in the application	1	High	Jayapradha B
Sprint-2		USN-3	As a user, youcan register in the application via Facebook	2	Low	Maimoon Shirin M
Sprint-1		USN-4	As a user, you can register in the application via Gmail	2	Medium	Kiruthika P
Sprint-1	Login	USN-5	As a user, you can login to the application by entering your emailand password	1	High	Banu Swetha R

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	29 Oct 2022	04 Nov 2022	20	03 Nov 2022
Sprint-2	20	4 Days	04 Oct 2022	08 Nov 2022	20	07 Nov 2022
Sprint-3	20	4 Days	08 Nov 2022	11 Nov 2022	20	10 Nov 2022
Sprint-4	20	4 Days	11 Nov 2022	14 Nov 2022	20	13 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points perday)

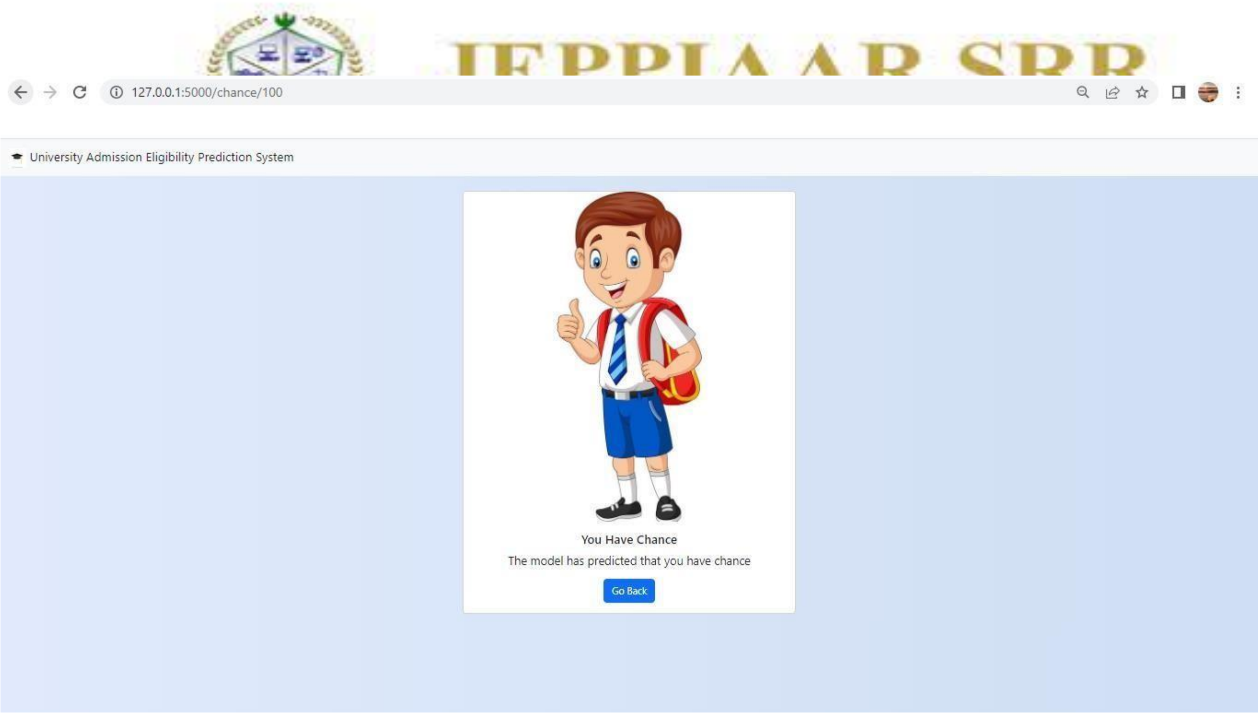
Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts canbe applied to any project containing measurable progress over time.

7. CODING & SOLUTIONING

7.1 Feature 1

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.



127.0.0.1:5000/chance/100

University Admission Eligibility Prediction System

You Have Chance

The model has predicted that you have chance

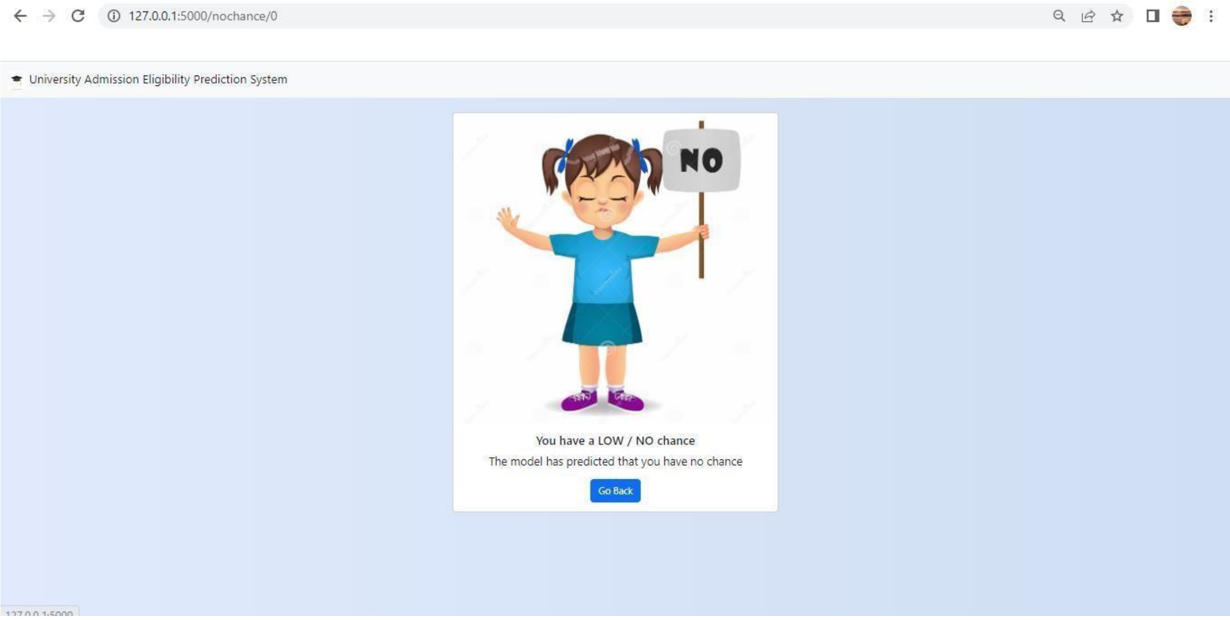
Source Code :

```
{% extends 'index.html' %}

{% block body %}

<div class="container text-center p-4">

    <div class="d-flex justify-content-center">

         <div class="card" style="width: 34rem;">

            <img src="..\static\img\chance.png" class="card-img-top" alt="...">

            <div class="card-body">

                <h5 class="card-title">You Have Chance</h5>
                <p class="card-text">The model has predicted that you have chance</p>

                <a href="/home" class="btn btn-primary">Go Back</a>

            </div>

        </div>

    </div>

</div>
```

```
{% endblock %}
```

7.2 Feature 2

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


127.0.0.1:5000/nochance/0
University Admission Eligibility Prediction System
NO
Go Back
127.0.0.1:5000

You have a LOW / NO chance

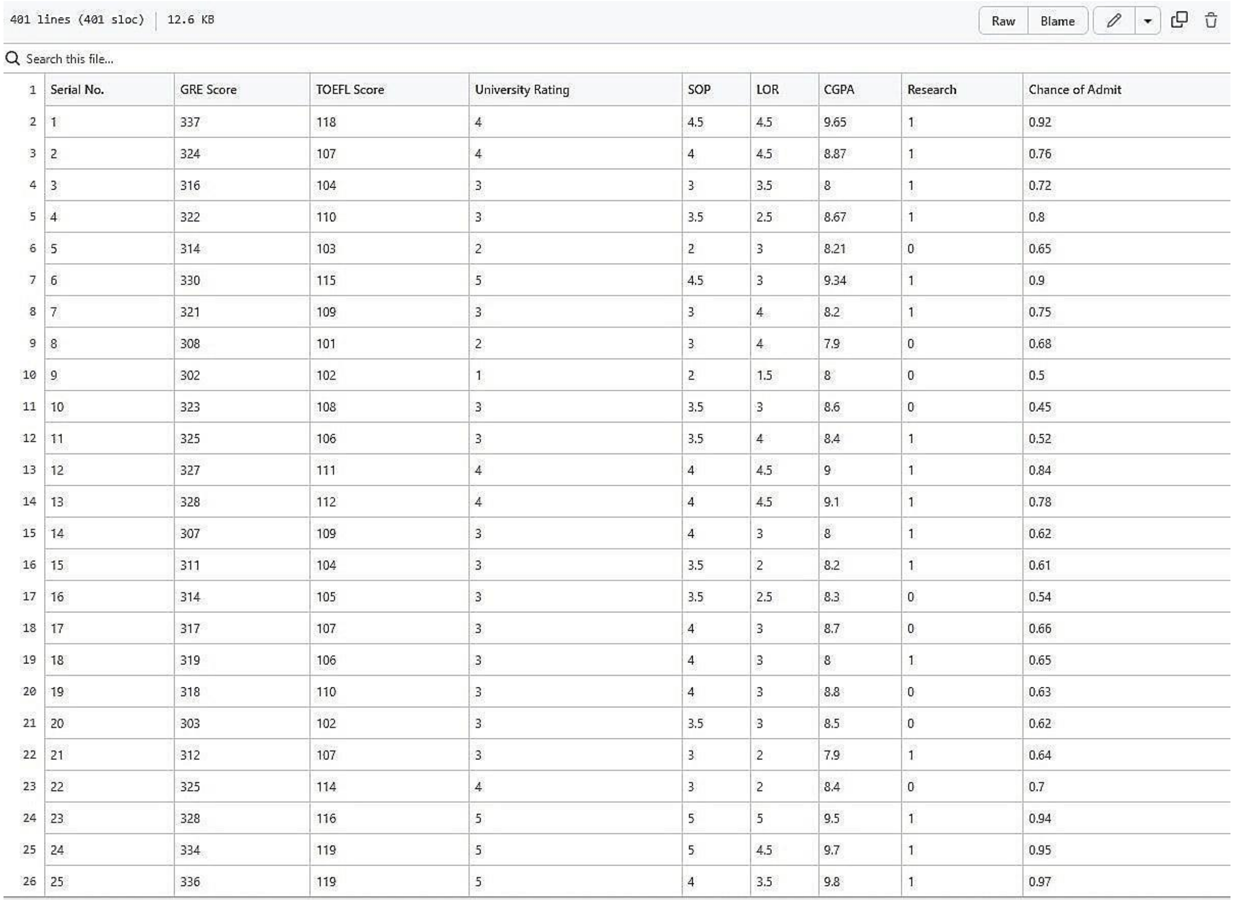
The model has predicted that you have no chance

Source Code:

```
{% extends 'index.html' %}

{% block body %}


<div class="container text-center p-4">

    <div class="d-flex justify-content-center">

        <div class="card" style="width: 34rem;">

            <img src="..\static\img\Nochance.jpg" class="card-img-top" alt="...">

            <div class="card-body">

                <h5 class="card-title">You have a LOW / NO chance</h5>

                 <p class="card-text">The model has predicted that you have no chance</p>
            <a href="/home" class="btn btn-primary">Go Back</a> </div>
        </div>
    </div>
</div>
{% endblock %}
```

7.3 Database Schema

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



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

8. TESTING

8.1 Test Cases

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.2 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

Resolution	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 Reproduced	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

Test Case Analysis

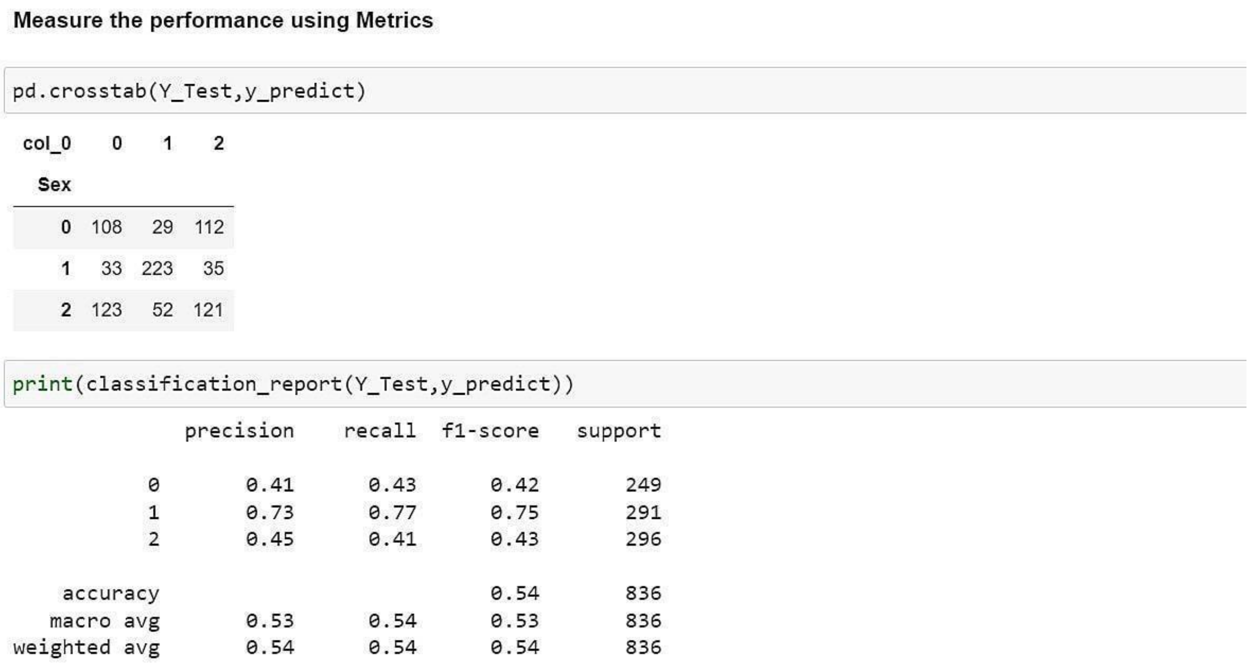
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.

RESULTS

9.1 Performance Metrics



Measure the performance using Metrics

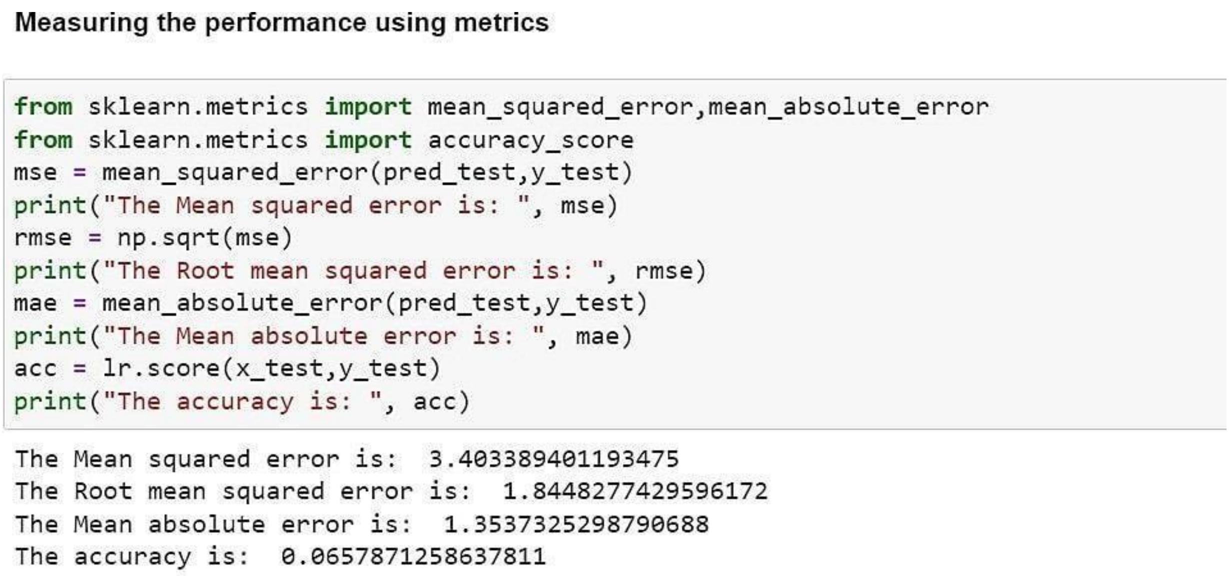
```
pd.crosstab(Y_Test,y_predict)
```

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

10.



Measuring the performance using metrics

```
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score
mse = mean_squared_error(pred_test,y_test)
print("The Mean squared error is: ", mse)
rmse = np.sqrt(mse)
print("The Root mean squared error is: ", rmse)
mae = mean_absolute_error(pred_test,y_test)
print("The Mean absolute error is: ", mae)
acc = lr.score(x_test,y_test)
print("The accuracy is: ", acc)
```

```
The Mean squared error is:  3.403389401193475
The Root mean squared error is:  1.8448277429596172
The Mean absolute error is:  1.3537325298790688
The accuracy is:  0.0657871258637811
```

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 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.

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.

12.

- 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 STATEMENTS

In [1]:

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

LOAD THE DATASET

In [2]:

```
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='T6FhPnWEPrnR91XKAfpiopbqTZ8j-gbLtjakMGexd6v0',
    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 = 'university-donotdelete-pr-1ijujvyruwxy5c' 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 object
 if not hasattr(body, "_iter_"): body._iter___= types.MethodType( __iter__, body )

data = pd.read_csv(body) data.head()
```

Out[2]:

	Serial No.	**GRE Score**	**TOEFL Score**	**University Rating**	**SOP**	**LOR**	**CGPA**	**Research**	**Chance of Admit**
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

In [3]:

```
data.drop(["Serial No."], axis=1, inplace=True)
```

In [4]:

```
data.describe()
```

Out[4]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [5]:

```
data.info()
```

Out[5]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
 #   Column             Non-Null Count Dtype

--  --------           ----------  ----
0  GRE Score         400 non-null   int64
1  TOEFL Score 400 non-null int64 2 University Rating 400 non-null
   int64
 3    SOP    400 non-null    float64
 4    LOR    400 non-null    float64
 5    CGPA  400 non-null    float64
 6    Research         400 non-null   int64
 7    Chance of Admit 400 non-null          float64 dtypes: float64(4), int64(4)
```

```
memory usage: 25.1 KB
```

In [6]:

```
data.isnull().sum()
```

Out[6]:

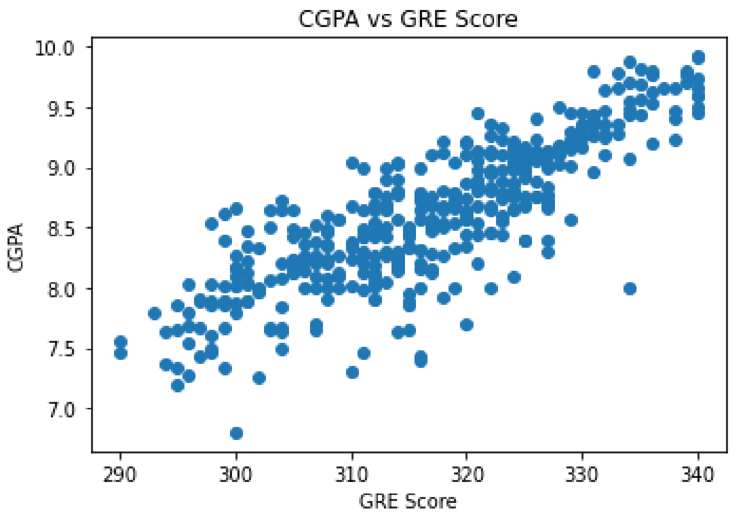
```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP 0 LOR          0
CGPA                 0
Research             0
Chance of Admit    0  dtype:
int64
```

VISUALIZATION

In [7]:

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

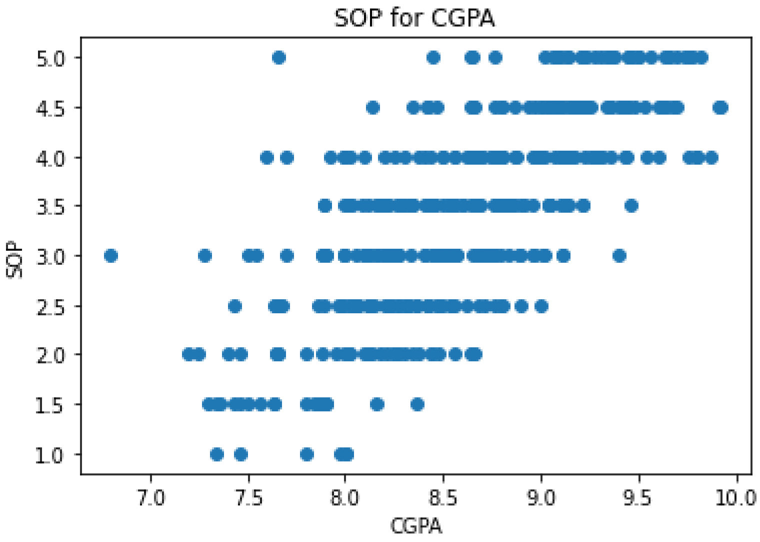
Out [7]:



In [8]:

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

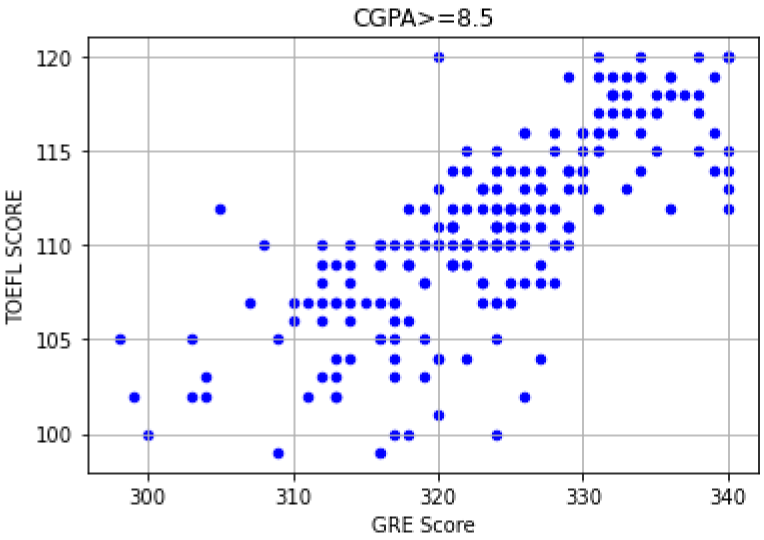
Out [9]:



In [9]:

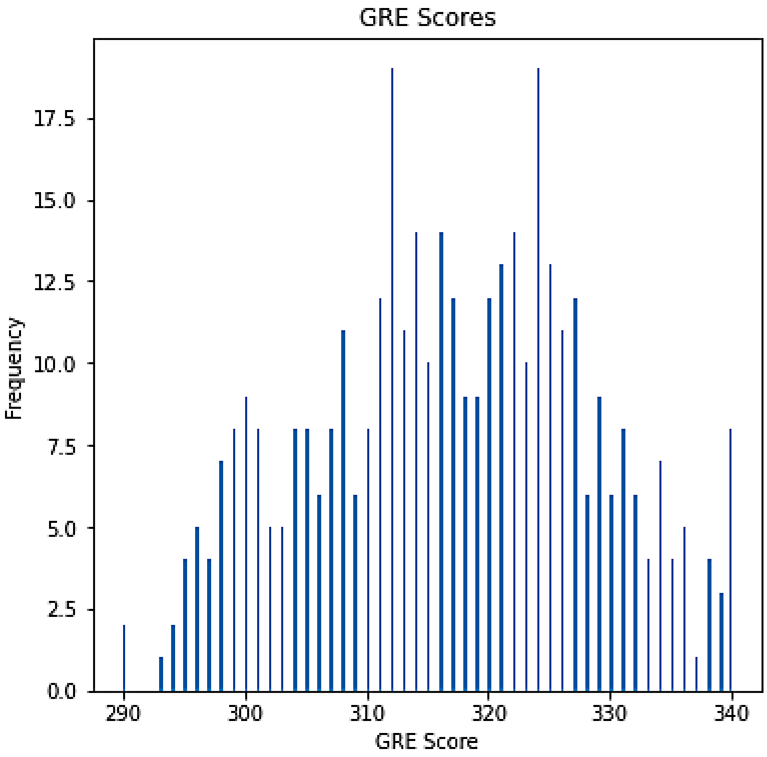
```
data[data.CGPA      >=       8.5].plot(kind='scatter',          x='GRE          Score',
        y='TOEFL Score',color="BLUE") plt.xlabel("GRE Score")
plt.ylabel("TOEFL SCORE")
plt.title("CGPA>=8.5")
plt.grid(True) plt.show()
```

Out [9]:



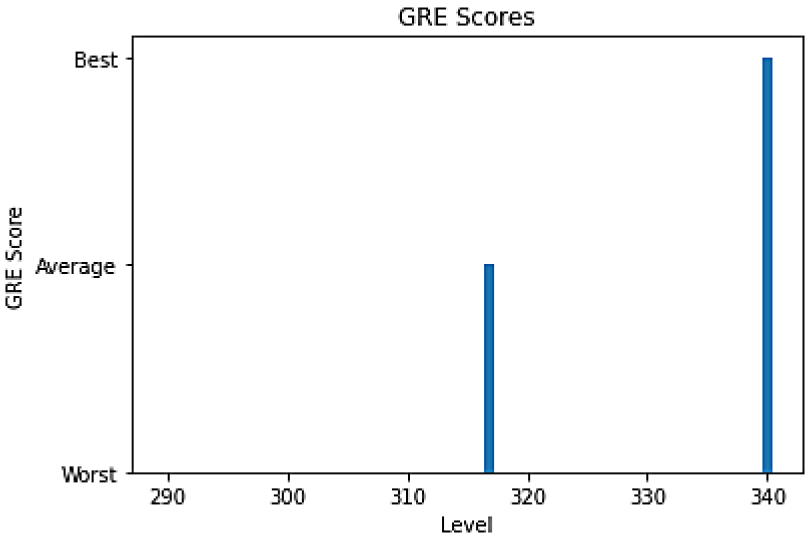
In [10]:

```
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[11]:

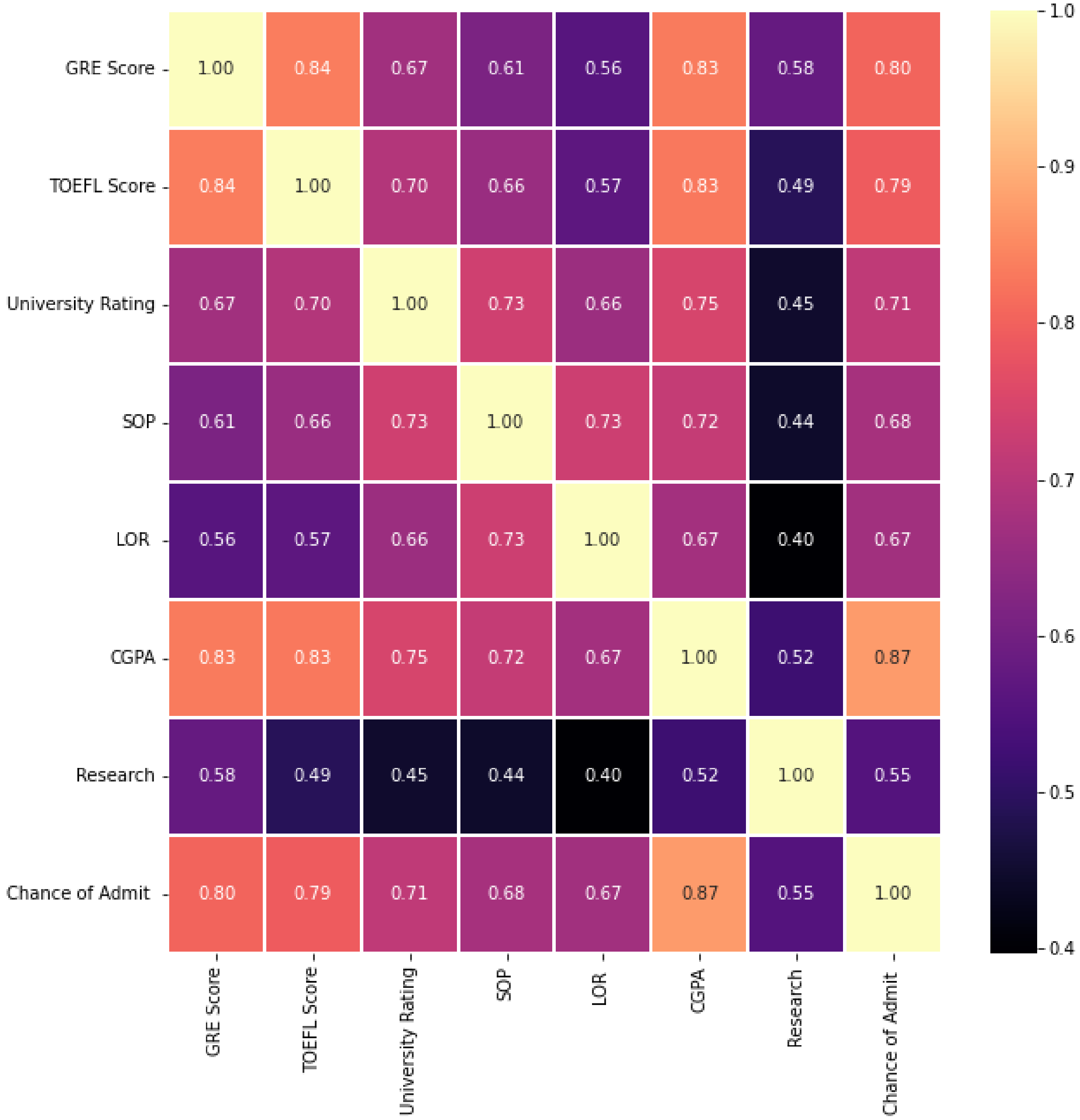
```
p = np.array([data["TOEFL Score"].min(),data["TOEFL
Score"].mean(),data["TOEFL Score"].max()]) r = ["Worst","Average","Best"]
plt.bar(p,r)

plt.title("TOEFL   Scores")
plt.xlabel("Level")
plt.ylabel("TOEFL
Score") plt.show()
```



In[13]:

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

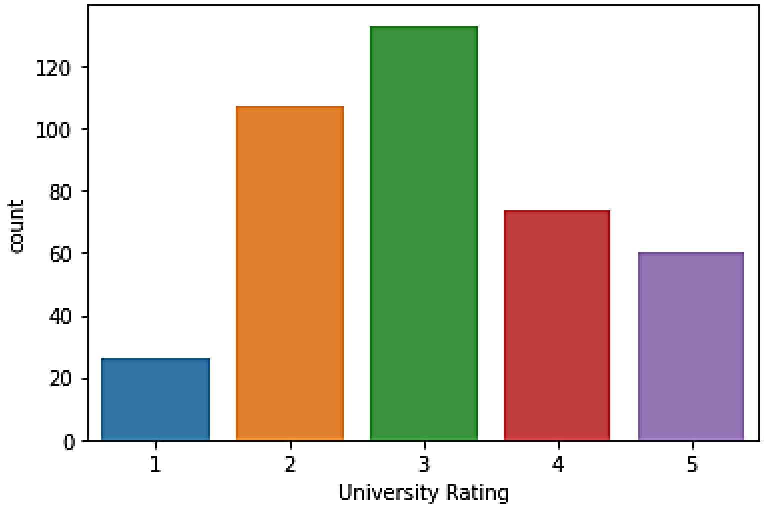
Out[13]:



In[14]:

data.Research.value_counts()
sns.countplot(x="University Rating",data=data) **Out[14]:**

```
<AxesSubplot:xlabel='University Rating', ylabel='count'>
```

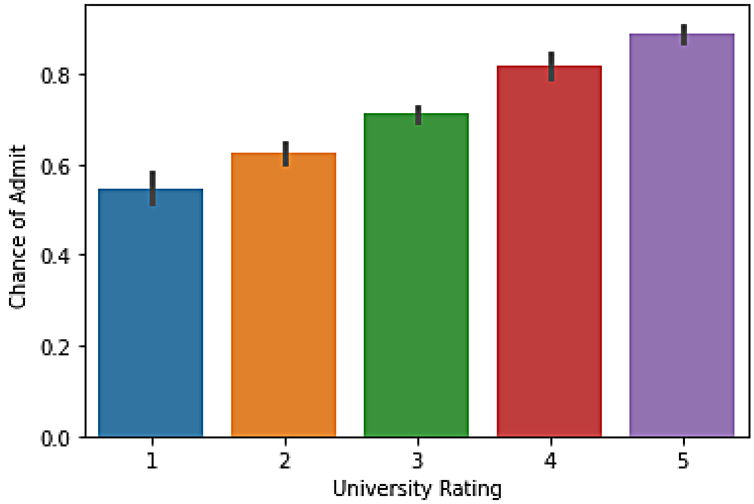


In[15]:

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

Out[15]:

```
<AxesSubplot:xlabel='University Rating', ylabel='Chance of Admit '>
```



TRAINING AND TESTING SPLIT

In [16]:

```
X=data.drop(['Chance of Admit '],axis=1) #input data_set y=data['Chance
of Admit '] #output labels
```

In [17]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15)
```

MODELING AND TRAINING

In [18]:

```
from sklearn.ensemble import
GradientBoostingRegressor rgr =
GradientBoostingRegressor() rgr.fit(X_train,y_train)
```

Out[18]:

```
GradientBoostingRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]:

```
rgr.score(X_test,y_test)
```

Out[19]:

0.7214021715194154 **In [20]:**

```
y_predict=rgr.predict(X_test)
```

In [21]:

```
from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
import numpy as np print('Mean Absolute Error:',
mean_absolute_error(y_test, y_predict)) print('Mean Squared Error:',
mean_squared_error(y_test, y_predict))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_predict)))
```

```
Mean Absolute Error: 0.061115035673946834
Mean Squared Error: 0.007194293635482686
Root Mean Squared Error: 0.08481918200196631
```

In [22]:

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

In [23]:

```
from sklearn.linear_model._logistic import LogisticRegression
lore = LogisticRegression(random_state=0, max_iter=1000) lr =
lore.fit(X_train, y_train)
```

In [24]:

```
y_pred = lr.predict(X_test)
```

In [25]:

```
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('Confussion Matrix:\n', confusion_matrix(y_test, y_pred))
```

Out [25]:

```
Accuracy Score: 0.9166666666666666
Recall Score: 1.0
ROC AUC Score: 0.7222222222222222 Confussion
Matrix:
[[ 4  5]
[ 0 51]]
```

SAVING THE MODEL

In [26]:

```
import pickle
```

In [27]:

```
pickle.dump(lr, open("university.pkl", "wb")) #logistic regression model
```

HOSTING THE MODEL

In [28]:

```
import pickle
```

In [29]:

```
lr = pickle.load(open("university.pkl", "rb")) #logistic regression model
```

In [30]:

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

Out [30]:

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-
3.9/lib/python3.9/site-packages (1.0.255)
Collecting ibm-watson-machine-learning
  Downloading ibm_watson_machine_learning-1.0.256-py3-none-any.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 22.1 MB/s eta 0:00:01
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: 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: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ibm-watson-machine-learning) (1.26.7)
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: 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: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site- packages
(from ibm-watson-machine-learning) (2.26.0)
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: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ibm-watson-machine-learning) (2022.9.24)
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: 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: 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: 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: 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: 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)
Installing collected packages: ibm-watson-machine-learning
  Attempting uninstall: ibm-watson-machine-learning
    Found existing installation: ibm-watson-machine-learning 1.0.255 Uninstalling
    ibm-watson-machine-learning-1.0.255:
      Successfully uninstalled ibm-watson-machine-learning-1.0.255
Successfully installed ibm-watson-machine-learning-1.0.256 Note:
you may need to restart the kernel to use updated packages.
```

In [31]:

```
from ibm_watson_machine_learning import APIClient import
json
```

In [32]:

```
uml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "poJ22ua6BCG9qY33B8fkgnz1bnP1f9DZqUlF9NkBM1bZ"
}        client        =
 APIClient(uml_credentials)
```

In [33]:

```
def guid_from_space_name(client, space_name):
    space =
    client.spaces.get_details() idr = []
    for i in space['resources']:
        idr.append(i['metadata']['id'])
    return idr
```

In [34]:

```
space_uid = guid_from_space_name(client, "university") print(space_uid[0])
4f0253e5-f162-4eec-84ba-72e01fb69ab9
```

In [35]:

```
client.set.default_space(space_uid[0])
```

Out[35]:

```
'SUCCESS'
```

In [36]:

```
client.software_specifications.list()
```

```
-------------------- ------------------------- ---
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     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 xgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-
60233c80306e base pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-
fb03b6f4fe12 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

-------------------- ------------------------- ---
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [37]:

```
import sklearn sklearn.
version
```

Out[37]:

```
'1.0.2'
```

In [38]:

```
MODEL_NAME = 'university'
DEPLOYMENT_NAME = 'uni'
DEMO_MODEL = lr
```

In [39]:

```
software_spec_uid = client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```
In [40]:

```
model_props = {
   client.repository.ModelMetaNames.NAME:                    MODEL_NAME,
   client.repository.ModelMetaNames.TYPE:           'scikit-learn_1.0            ',
   client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:
   software_spec_uid
}
```
In [41]:

```
model_details = client.repository.store_model( model
   = DEMO_MODEL,
   meta_props = model_props,
   training_data   =   X_train,
   training_target = y_train
)
model_details
```

Out[41]:

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'Chance of Admit ',
  'schemas': {'input': [{'fields': [{'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': '2022-11-03T05:20:49.371Z',
  'id': '566cfcae-49ab-4bd3-b5df-abc981fa27b9',
  'modified_at': '2022-11-03T05:20:51.730Z',
  'name': 'university',
  'owner': 'IBMid-6630041JHH',
  'resource_key': 'a61934d2-41d0-413d-9f54-49589e7c7741',
'space_id': '4f0253e5-f162-4eec-84ba-72e01fb69ab9'},
'system': {'warnings': []}}
```

In [42]:

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

Out[42]:

```
'566cfcae-49ab-4bd3-b5df-abc981fa27b9'
```

In [43]:

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

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

Out [43]:

```
##############################################################################
#########
Synchronous deployment creation for uid: '566cfcae-49ab-4bd3-b5df-abc981fa27b9' 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='28aea4f7-0bec-4310-82bf-
06e502d2cd4d'

------------------------------------------------------------------
```

HTML CODES

Uploading Html codes

Chance.html

```
{% extends 'index.html' %}

{% block body %}

<div class="container text-center p-4">

    <div class="d-flex justify-content-center">
```

```
          <div class="card" style="width: 34rem;">
            <img src="..\static\img\chance.png" class="card-img-top" alt="...">

            <div class="card-body">

                <h5 class="card-title">You Have Chance</h5>

                <p class="card-text">The model has predicted that you have chance</p>

                <a href="/home" class="btn btn-primary">Go Back</a>

            </div>

        </div>

    </div>

</div>

{% endblock %}
```

Demo2.html

```
{% extends 'index.html' %}

{% block body %}

    <div class="p-4">

       <div class="row mb-3">

            <div class="col-4">

                <h2 class="text-responsive-h">

                    Enter your details and get probability of your admission

                </h2>

                <p class="text-responsive">

                       Students are often worried about their chances of admission to University. The aim
of this project is to help students in shortlisting universities with their profiles. The predicted output
gives them a fair idea about their admission chances in a particular university. This analysis should
also help students who are currently preparing or will be preparing to get a better idea.
```

```
    </p>
    <div class="d-flex justify-content-right">
        <img src="../static/img/anime.png" class="card-img-top" alt="..." />
    </div>
    </div>
<div class="col-8">
    <div class="card p-2 ms-2 my-2">
        <div class="card-body">
            <h5 class="card-title pb-4">
                Enter the details
            </h5>
            <form action="/" method="post" id="theForm">
                <div class="row mb-3">
                    <label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="gre" name="gre" min="250" max="340" required>
                    </div>
                </div>
                <div class="row mb-3">
                    <label for="tofel" class="col-lg-2 col-form-label">TOFEL Score:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="tofel" name="tofel" min="50" max="120" required>
                    </div>
                </div>
                <div class="row mb-3">
                    <label for="university_rating" class="col-lg-2 col-form-label">University Rating:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="university_rating"
```

```
step="0.01" name="university_rating" min="1" max="5" required> </div>
            </div>
            <div class="row mb-3">
              <label for="sop" class="col-lg-2 col-form-label">SOP:</label>
              <div class="col-lg-10">
                <input type="number" class="form-control" id="sop" name="sop"
step="0.01" min="1" max="5" required>
              </div>
            </div>
            <div class="row mb-3">
              <label for="lor" class="col-lg-2 col-form-label">LOR:</label>
              <div class="col-lg-10">
                <input type="number" class="form-control" id="lor" name="lor"
step="0.01" min="1" max="5" required>
              </div>
            </div>
            <div class="row mb-3">
              <label for="cgpa" class="col-lg-2 col-form-label">CGPA:</label>
              <div class="col-lg-10">
                <input type="number" class="form-control" id="cgpa" name="cgpa"
step="0.01" min="5" max="10" required>
              </div>
            </div>
            <fieldset class="row mb-3">
              <legend class="col-form-label col-sm-2 pt-0">Research:</legend>
              <div class="col-sm-10">
                <div class="form-check">
                  <input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios1" value="1">
                  <label class="form-check-label" for="yes_no_radio">
                  Yes
```

```
                                    </label>
                                </div>
                                <div class="form-check">
                                    <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios2" value="0" checked>
                                    <label class="form-check-label" for="yes_no_radio">
                                    No
                                    </label>
                                </div>
                            </div>
                        </fieldset>
                        <div class="row lg-3">
                            <div class="col-lg-2 mb-2 me-3">
                                <button type="submit" class="btn btn-primary" id="button">Predict</button>
                            </div>
                            <div class="col-lg-2" id="spinner">
                                <div class="spinner-border text-primary m-1" role="status">
                                    <span class="visually-hidden">Loading...</span>
                                </div>
                                <div class="spinner-grow text-primary m-1" role="status">
                                    <span class="visually-hidden">Loading...</span>
                                </div>
                            </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
{% endblock %}
```

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">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="..\static\img\hat1.png" alt="Logo" width="30" height="24" class="d-inline-block align-text-top ">
                University Admission Eligibility Prediction System </a>
        </div>
    </nav>
    {% block body %}
    <h1> Index Page </h1>

    {% endblock %}

    <script            src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>

</body>

</html>
```

Nochance.html

```
{% extends 'index.html' %}

{% block body %}


<div class="container text-center p-4">

    <div class="d-flex justify-content-center">

        <div class="card" style="width: 34rem;">

            <img src="..\static\img\Nochance.jpg" class="card-img-top" alt="...">
            <div class="card-body">

                <h5 class="card-title">You have a LOW / NO chance</h5>

                 <p class="card-text">The model has predicted that you have no chance</p>

                <a href="/home" class="btn btn-primary">Go Back</a> </div>

        </div>

    </div>

</div>

{% endblock %}
```

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;
    padding:
    0; border:
    0;
} body
{
     font: 62.5%/1.5 "Lucida Grande", "Lucida Sans", Tahoma, Verdana, sans-serif; background:
     #e0eafc;
     background: -webkit-linear-gradient(to right, #e0eafc,
     #cfdef3); background: linear-gradient(to right, #e0eafc,
     #cfdef3); color: #000000; text-align:center;
  }


  h1 {
     font-size: 2.2em;
  }

  h2    {    font-size:
     2.0em;
  }

  h4    {    font-size:
     1.6em;
  }
 p { font-size:
1.2em;
  }
 input.text { padding: 3px;
border: 1px solid #999999;
  }

  img {
     max-width: auto; height:
     auto;
  }

  .text-responsive { font-size: calc(50% +
     0.6vw + 0.6vh);
```

```
}
.text-responsive-h { font-size: calc(80%

   + 0.6vw + 0.6vh);

}
```

Footer

REQUIRED IMAGES IN THE HTML CODES :

NO

PYTHON CODE

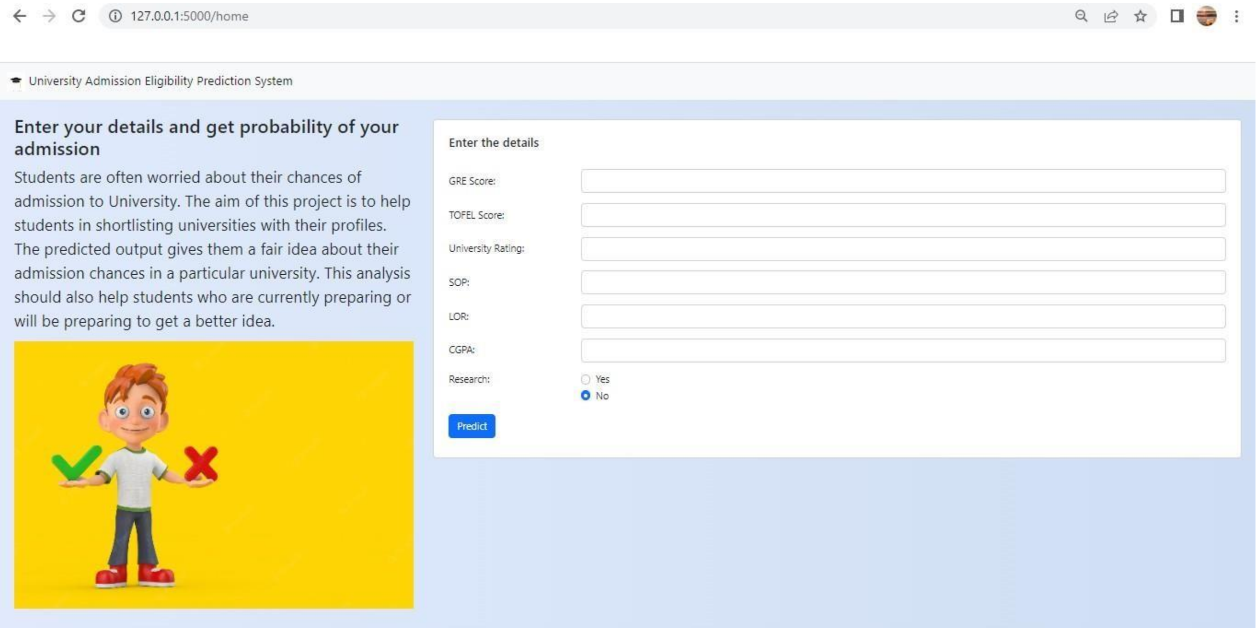
App.py

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


  app = Flask(_name__)


  @app.route("/", methods = ['POST', 'GET']) def
  index():
      if request.method == 'POST':
```

```
arr = [] for i in
request.form: val =
request.form[i] if val
== '':
        return redirect(url_for("demo2"))
    arr.append(float(val))

# deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
API_KEY = "poJ22ua6BCG9qY33B8fkgnz1bnP1f9DZqUlF9NkBM1bZ"
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}
payload_scoring = {
    "input_data": [{"fields":[ 'GRE Score',
                              'TOEFL Score',
                              'University Rating',
                              'SOP',
                              'LOR ',
                              'CGPA',
                              'Research'],
                    "values": [arr]
                    }]
                }

response_scoring = requests.post(
    'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/28aea4f7-0bec-4310-82bf-
06e502d2cd4d/predictions?version=2022-11-03', json=payload_scoring, headers=header
```

```
         ).json()

result = response_scoring['predictions'][0]['values']

if result[0][0] > 0.5:
             return redirect(url_for('chance', percent=result[0][0]*100))
         else:          return          redirect(url_for('no_chance',
             percent=result[0][0]*100))
      else:
         return redirect(url_for("demo2"))
  @app.route("/home") def
  demo2():
      return render_template("demo2.html")


  @app.route("/chance/<percent>")
  def chance(percent):
      return render_template("chance.html", content=[percent])


  @app.route("/nochance/<percent>")
  def no_chance(percent):
      return render_template("noChance.html", content=[percent])


  @app.route('/<path:path>') def
  catch_all():
      return redirect(url_for("demo2"))


  if __name___== "_main_":
      app.run()
```

OUTPUT IMAGES:



127.0.0.1:5000/home

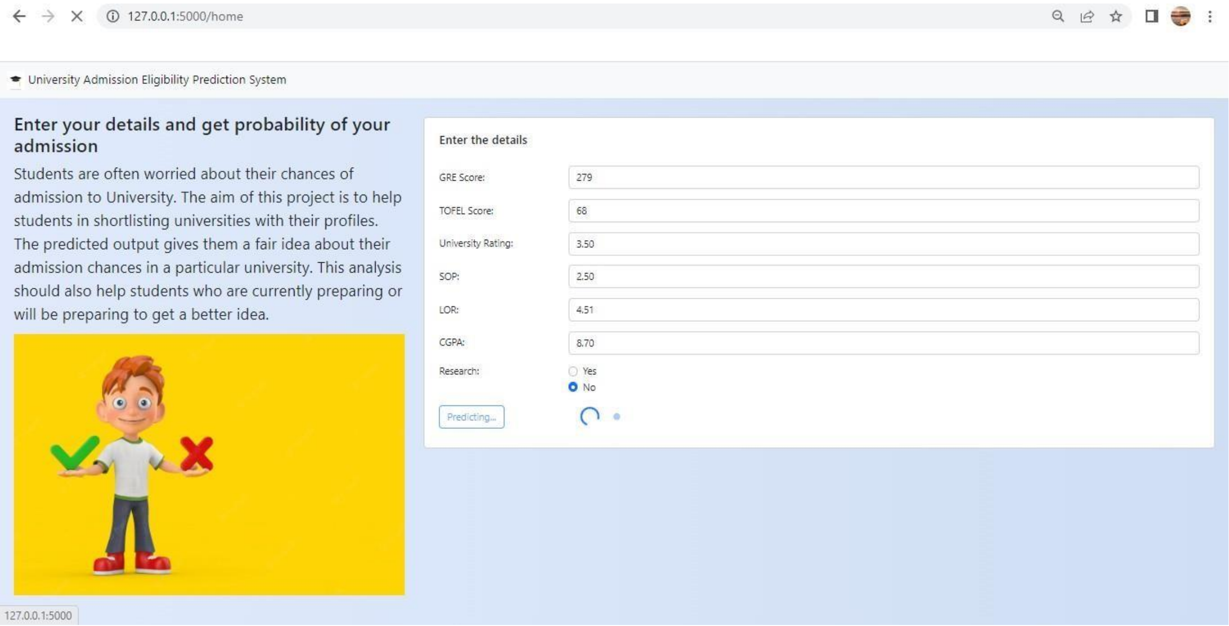
University Admission Eligibility Prediction System

Enter your details and get probability of your admission

Students are often worried about their chances of admission to University. The aim of this project is to help students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

Enter the details

GRE Score:	
TOFEL Score:	
University Rating:	
SOP:	
LOR:	
CGPA:	
Research:	○ Yes ● No



127.0.0.1:5000/home

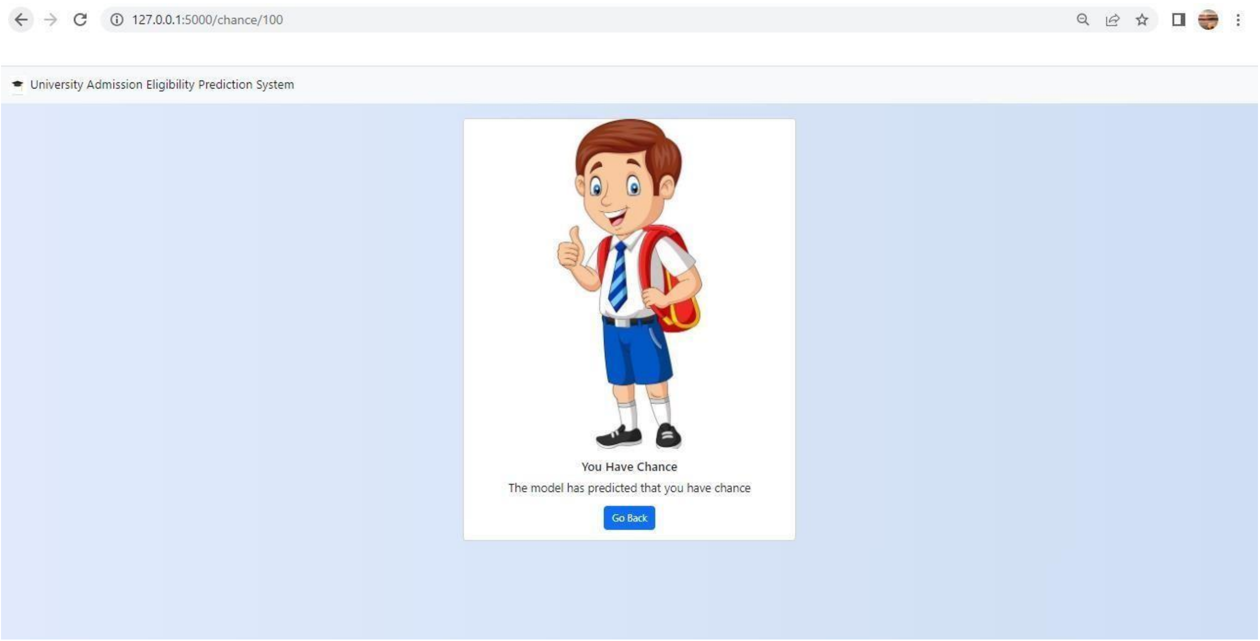
University Admission Eligibility Prediction System

Enter your details and get probability of your admission

Students are often worried about their chances of admission to University. The aim of this project is to help students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

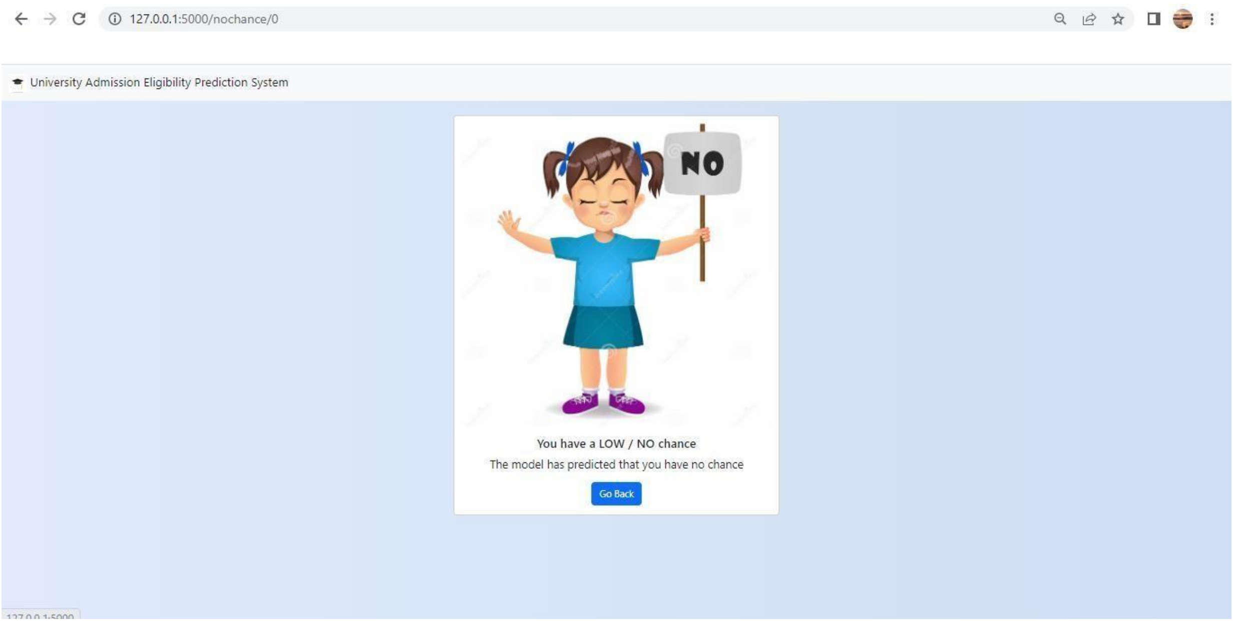
Enter the details

GRE Score:	279
TOFEL Score:	68
University Rating:	3.50
SOP:	2.50
LOR:	4.51
CGPA:	8.70
Research:	○ Yes ● No


127.0.0.1:5000/chance/100
University Admission Eligibility Prediction System
Go Back

You Have Chance

The model has predicted that you have chance


127.0.0.1:5000/nochance/0
University Admission Eligibility Prediction System
NO
Go Back

You have a LOW / NO chance

The model has predicted that you have no chance

13.2 GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-2118-1658459214