UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

USING APPLIED DATA SCIENCE

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Team ID: PNT2022TMID34608

Aysha Nivera A	962219104029
Amala Selciya T L	962219104012
Abina R	962219104003
Aswathy P M	962219104026
Ancy N	962219104501

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ST. XAVIER'S CATHOLIC COLLEGE OF ENGINEERING

ANNA UNIVERSITY: CHENNAI 600025

NOV-2022

ST. XAVIER'S CATHOLIC COLLEGE OF ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"UNIVERSITY ADMIT ELIGIBILITY PREDICTOR"** is the bonafide record work done by **Ms AYSHA NIVERA A** (**962219104029**)**, Ms AMALA SELCIYA T L (962219104012), Ms ABINA R (962219104003), Ms ASWATHY P M (962219104026)** and **MS ANCY N R**(**962219104501**) for **IBM-NALAIYATHIRAN** in **VII** semester of **B.E.,** degree course **in Computer Science and Engineering** branch during the academic year of 2022 - 2023.

Staff-In charge **Evaluator**

Mrs. S.T. Sheriba **Mr. S. John Pimo**

Head of the Department

Mrs. P.R. Sheeba Rani

ACKNOWLEDGEMENT

We express our breathless thanks to our **Dr.J.Maheswaran, M.E., Ph.D.,** the principal, St. Xavier's catholic College of Engineering, Chunkankadai, for giving constant motivation in succeeding in our goal. We acknowledge our sincere thanks to Head of the Department **Mrs. P.R. SHEEBA RANI** for giving us valuable suggestion and help towards us throughout this Project. We are highly grateful to thank our Project coordinator **Mrs. S.T. Sheriba** And Our Project Evaluator **Mr. S. John Pimo** Department of Computer Science and Engineering, St.Xavier's catholic college of Engineering and Technology for the coordinating us throughout this Project. We are very much indebted to thank all the faculty members of Department of Computer science and Engineering in our Institute, for their excellent moral support and suggestions to complete our Project work successfully. Finally our acknowledgment does our parents, sisters and friends those who had extended their excellent support and ideas to make our Project a pledge one.

Aysha Nivera A

Amala Selciya T L

Abina R

Aswathy P M

Ancy N

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

1. Graduate Record Examination 1 (GRE) score. The score consists of 340 foci.
2. English as a Foreign Language (TOEFL) test score. It consists of 120 priority areas.
3. UniRating. Shows the position of colleges offering bachelor's degrees among various colleges. Your score will be out of 5.
4. 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.
5. 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.
6. Undergraduate GPA (CGPA) from 10.

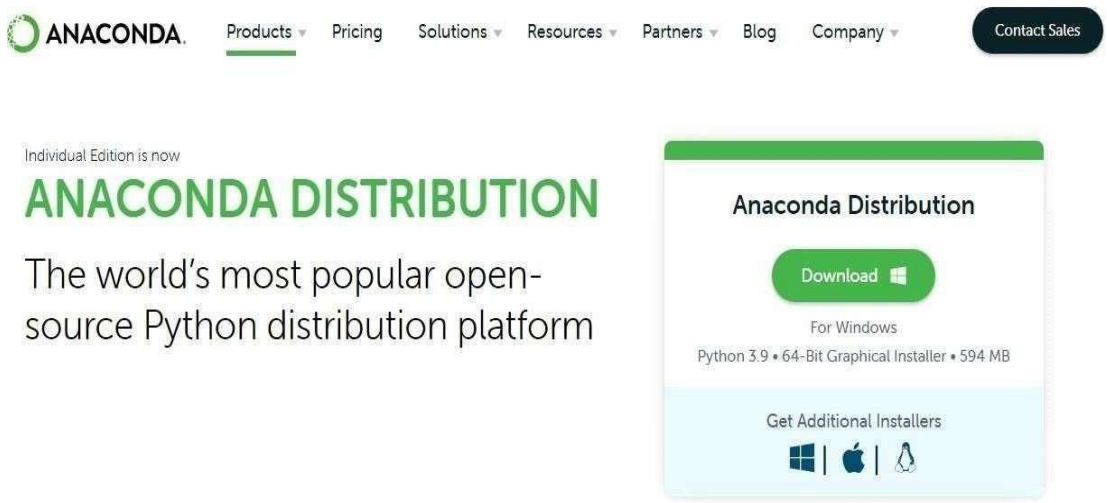
Research experience (either 0 or 1) that could support the application, such as distributing research papers at conferences or filling out as a right-hand exam for university faculty. One ward variable can be anticipated which is possibility of affirmation, that is as per the input given will be going from 0 to 1

PROJECT OVERVIEW

Anaconda Installation:

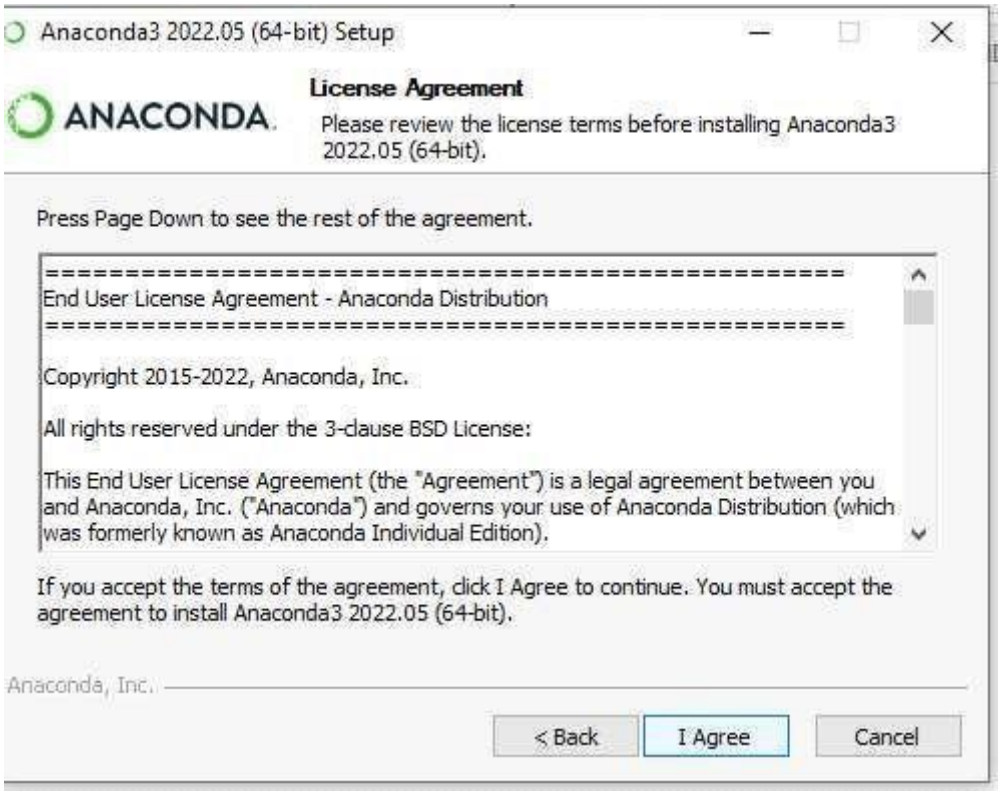
Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data science packages suitable for Windows, Linux, and macOS. Developed and maintained by Anaconda. Founded in 2012 by Peter 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.

Step 1: Download and Anaconda

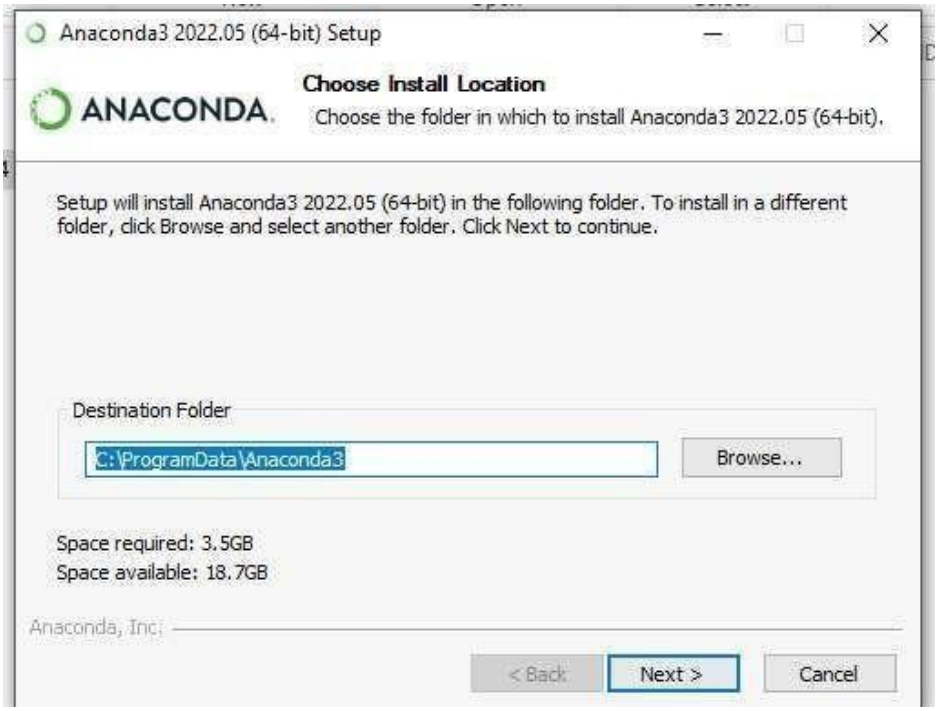


Step 2: Install the Anaconda

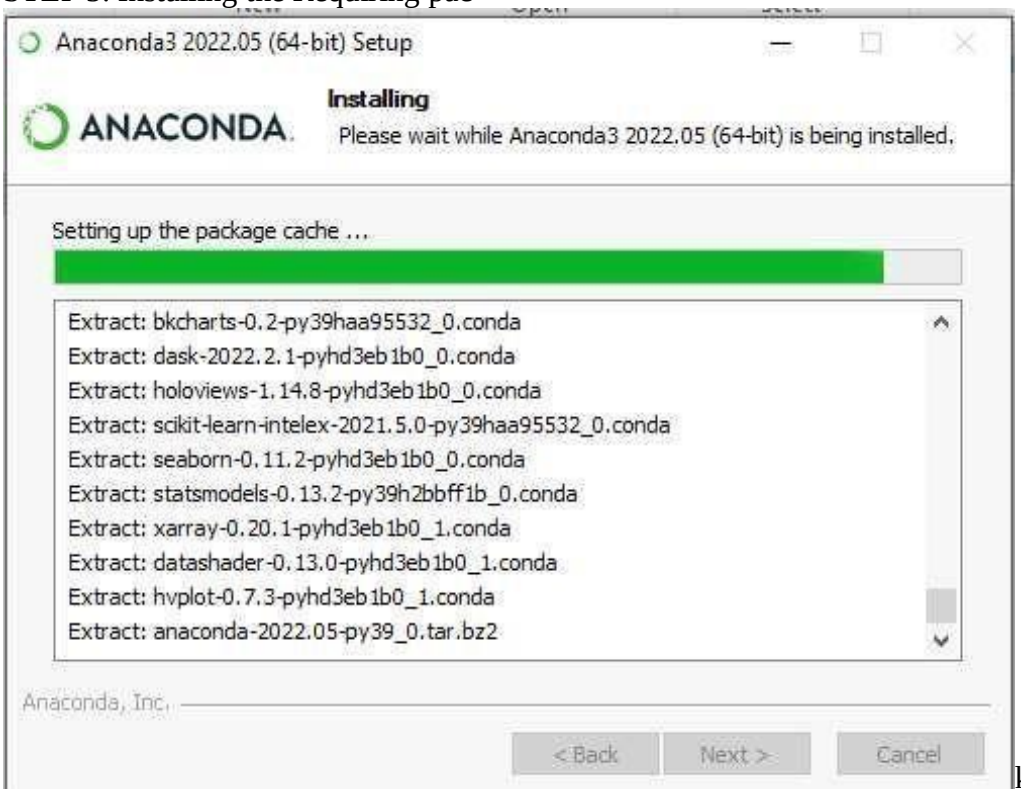
STEP 3: Click I Agree



STEP 4: Choose the Location

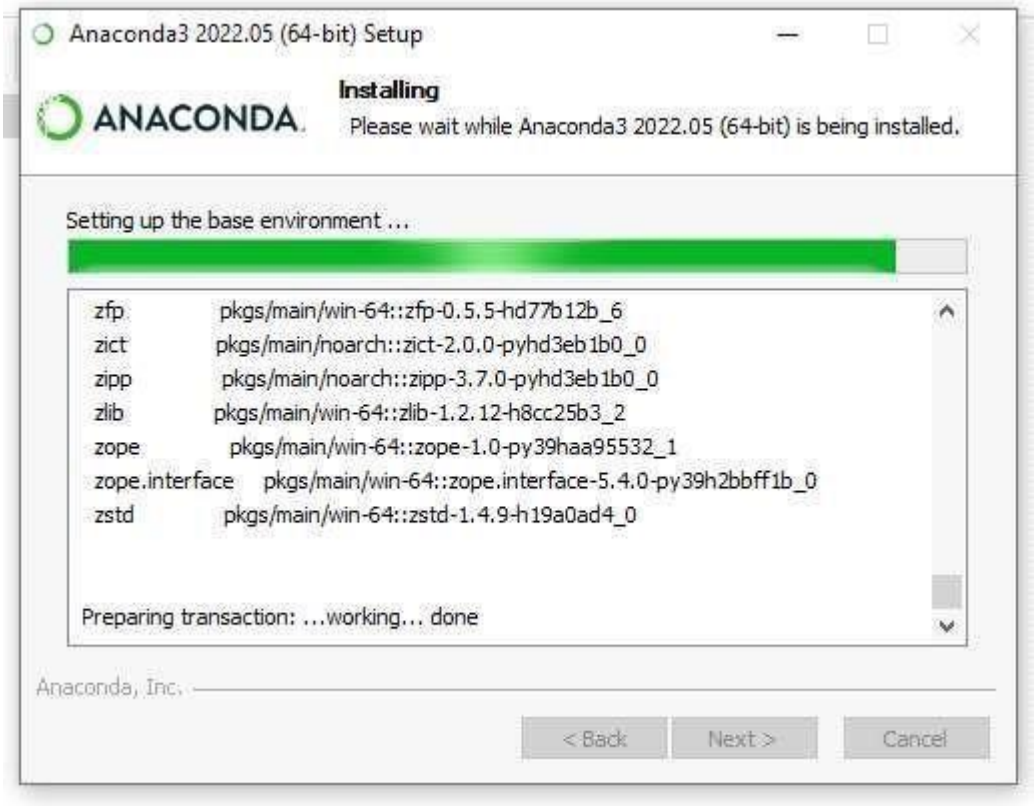


STEP 5: Installing the Requiring pac

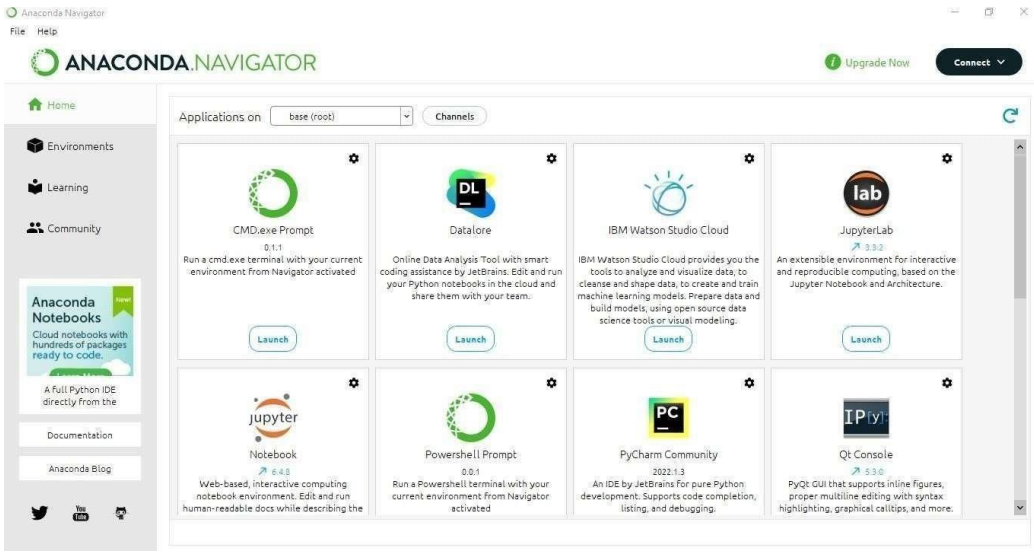


kages

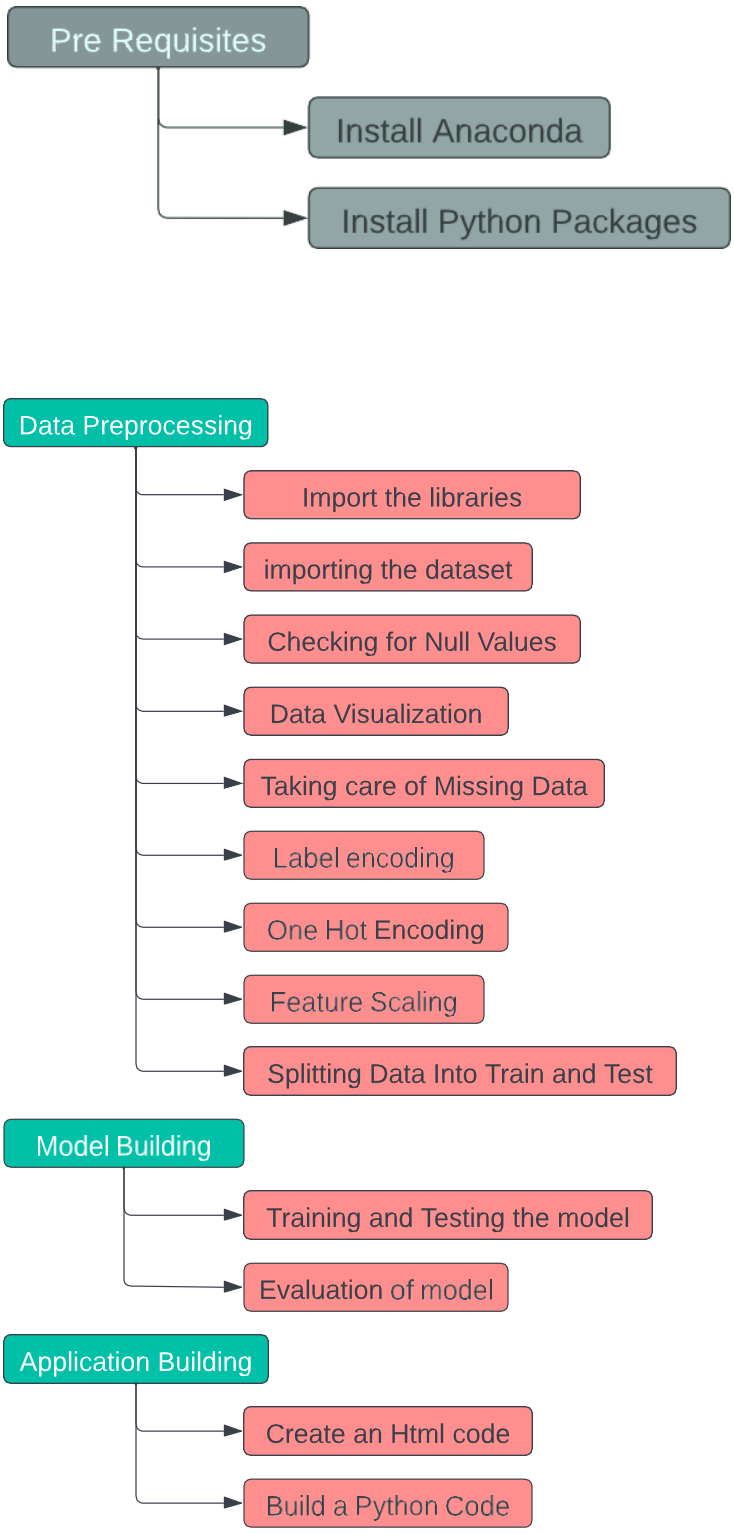
STEP 6: Setting up the base environment



STEP 7: Successfully Installed .



PROJECT FLOW



PROJECT OBJECTIVES

This project University Admission Predictor System is web based application in which students can enter their marks details for prediction the admission in colleges. Using this software, the entrance seat allotment became easier and can be implemented using system. The students only have to enter their marks of GRE, TOFEL, UNIVERSITY RANK, etc. This will not make the admission process easy but also minimizes stress for students. The main objective of our system is to make the right choice of colleges. College admission predictor is a helpful to many students.The students not only filling the form but the students give an idea about their future college by calculating their cut off. When the students come from rural places, they find it hard to go along with the formal procedures. So, this application helps them a lot and cases out their fear. Whatever maybe their scores, this application helps to find the best colleges. Hence, our project will help the students to get the list of colleges in which they could get the admission at the click of a button.

DATA PRE-PROCESSING

Importing the Libraries:

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

1.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.

2.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.

3.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 commandData=
pd.read_csv(r"File_location/filename.csv")

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

Our Dataset Admission_Predict contains
following Columns1.Serial No.

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

Handling Missing Values:

After loading it is important to check the complete information of data as it can indicate many ofthe 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

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

```
In [7]: data.isnull().sum()

Out[7]: GRE Score            0
        TOEFL Score          0
        University Rating    0
        SOP                  0
        LOR                  0
        CGPA                 0
        Research             0
        Chance of Admit      0
        dtype: int64
```

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

b. You can also check the number of null values present in the columns by the usingisnull().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.

a. To read the columns, we will useiloc of pandas (used to fix theindexes 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)
```

a. You can see in x we have 1991 rows with 7 columns and y has 1 column with the samenumber 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 datasetinto 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.

b. Train Dataset:Used to fit the machine learning model.

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

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

a. x_train
b. x_test
c. y_train
d. y_test .

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

e. 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
f. 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.

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 andimplementation of the target system in an object-oriented language.

2. LITERATURE SURVEY

Existing Problem

It’s almost admission season and I’ve couple of friends who are in panic mode waiting for a callfrom the universities they’ve applied at.
This made me think — How can we predict whether a student will get an admit or not? What arethe parameters for selection? Can it be mathematically expressed?
All of these questions started popping up. This is the main existing problem.

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 WCE2011.

> 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:

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

Problem Statement Definition

Problem Statement(PS)	I am (Customer)	I'm trying to	But	Because	Which makes mefel
PS-1	Student	I am looking university forlong 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 tofind the university.
PS-2	Student	I am Searching University on my phone	It is difficultto find and it takes long time	It is difficult tosearch in offline	Exhausted

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

Abstract:

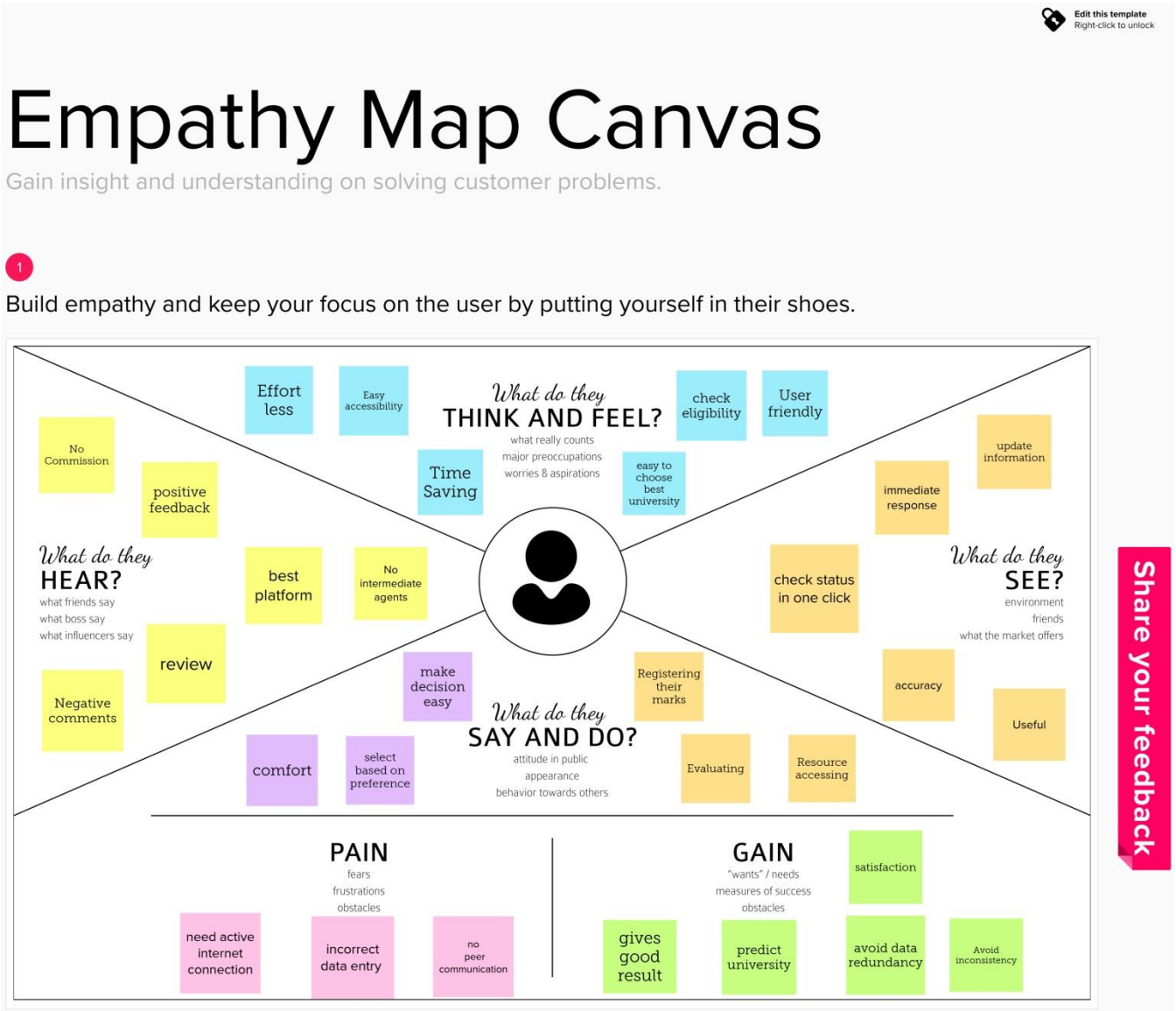
This system was originally developed only for Engineering College Admissions in Maharashtra,India but can essentially be adapted for other streams too. The purpose of it is to build a system to predict the users chances for getting into a certain college.

3. IDEATION AND PROPOSED SOLUTION

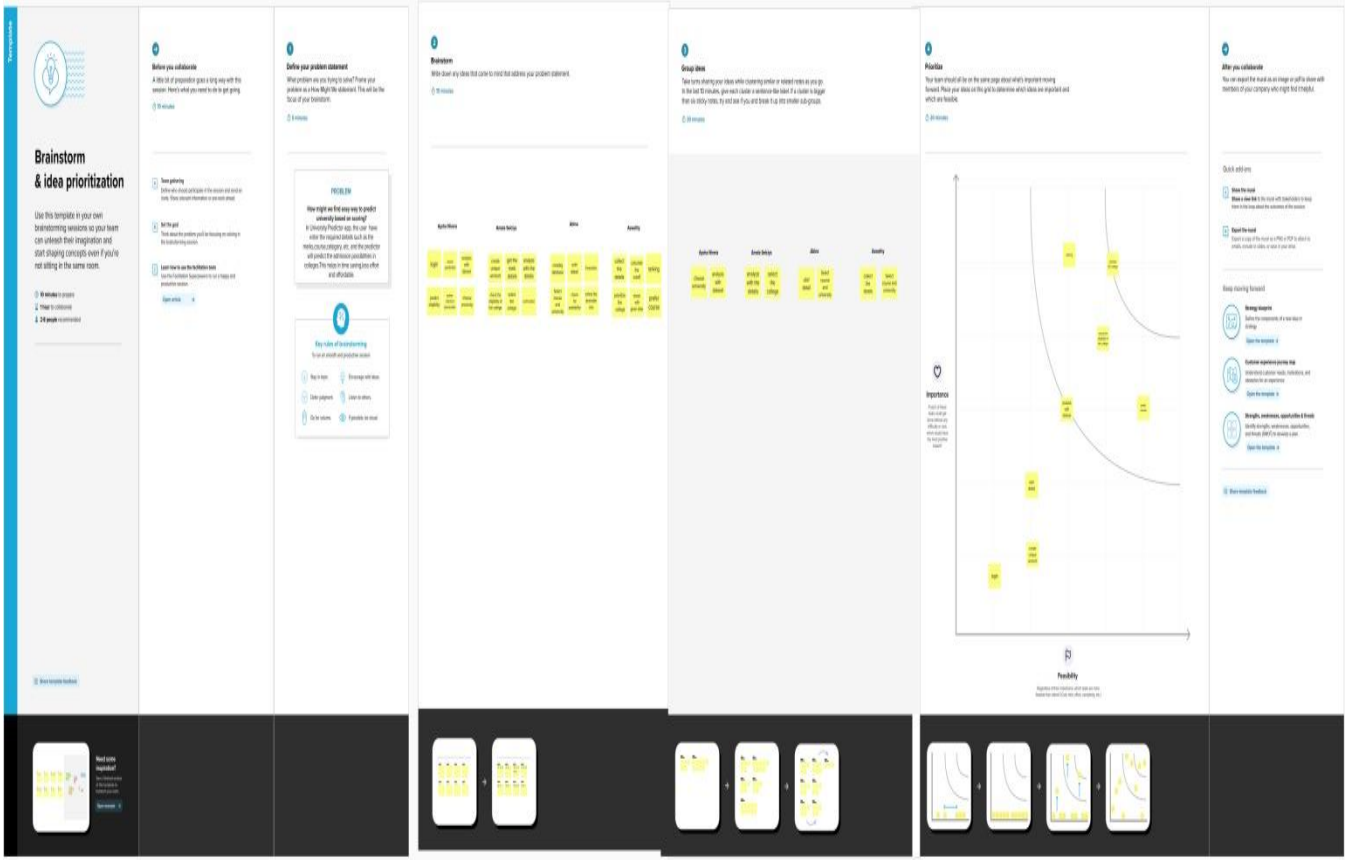
Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process. In thisproject the ideation phase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

1. Empathy Map
2. Brainstorming
3. Proposed Solution
4. Problem Solution Fit

3.1 EMPATHY MAP



.3.2 BrainStorming & Ideation

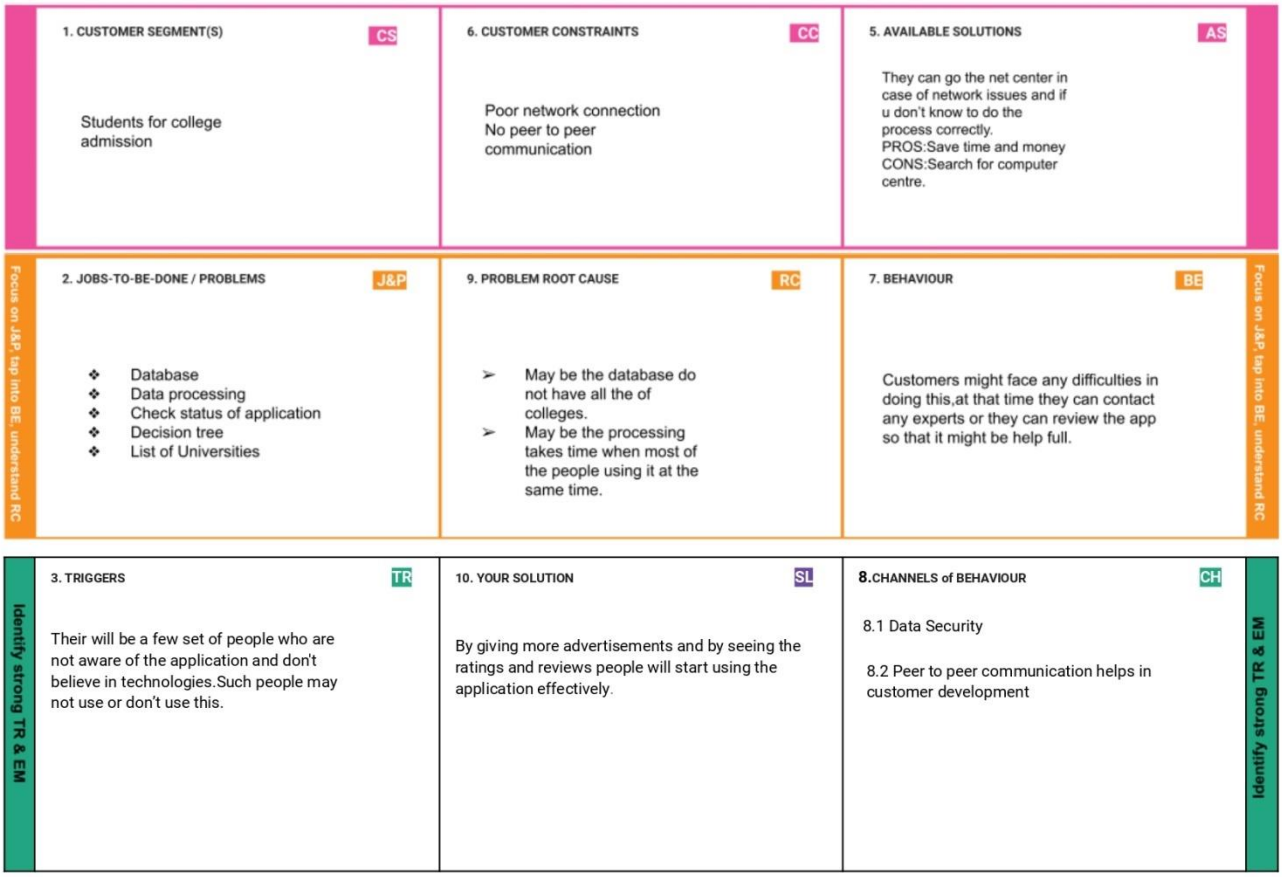


Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Predict the chances of admission in the universities
2.	Idea / Solution description	Create a web application that predicts the user's possibility of admission in the universities of their choice for PG studies.
3.	Novelty / Uniqueness	Unique Dataset which is required for the University prediction has been used in the project. This can be implemented as a new method.
4.	Social Impact / Customer Satisfaction	This project will help the student in making decisions for choosing a right college and satisfy the customer because they can check their eligibility in a single click by giving the user data anywhere at any time instead of wasting a lot of time and money.
5.	Business Model (Revenue Model)	It will help students to make correct decisions for choosing a right college. This project helps the colleges and Universities to advertise and to build their brand and also to get more chances of admission.
6.	Scalability of the Solution	More than 2000 students can use this prediction by using machine learning.

Problem Solution fit

Project Title: University Admit Eligibility Predictor | Project Design Phase-I - Solution Fit Template | Team ID: PNT2022TMID34608



4.Requirment Analysis

Functional Requirment

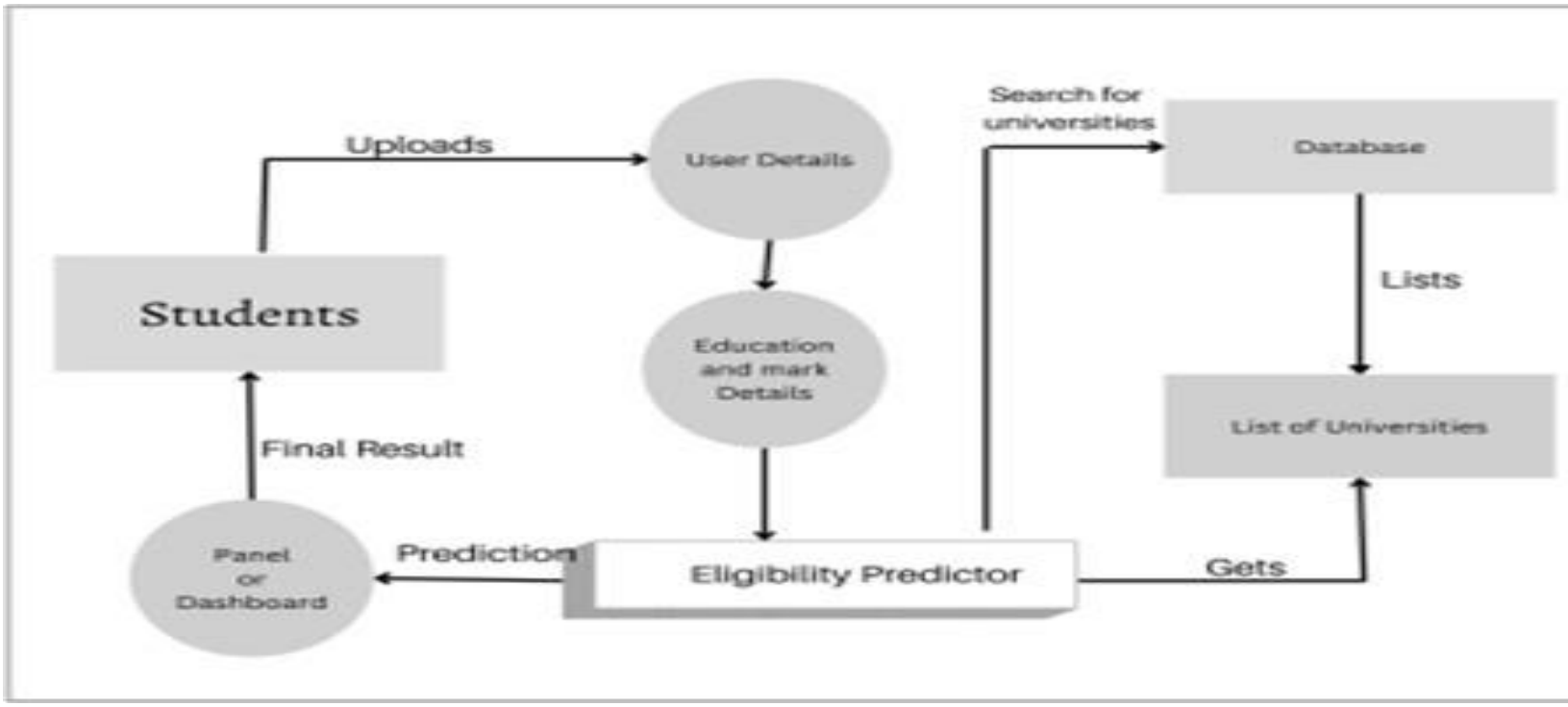
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Submit the documents GRE and TOEFL score sheetCurriculum Vitae (CV) Statement of Purpose (SoP)
FR-4	User Requirements	Upload all the relevant documents in the appropriate location in the website The list of all possible university for the candidate would be displayed based on the information
FR-5	Prediction	Prediction is done based on theresult of evaluation, the List of Universities for which the students are eligible toapply will be displayed.

Non Functional Requirment

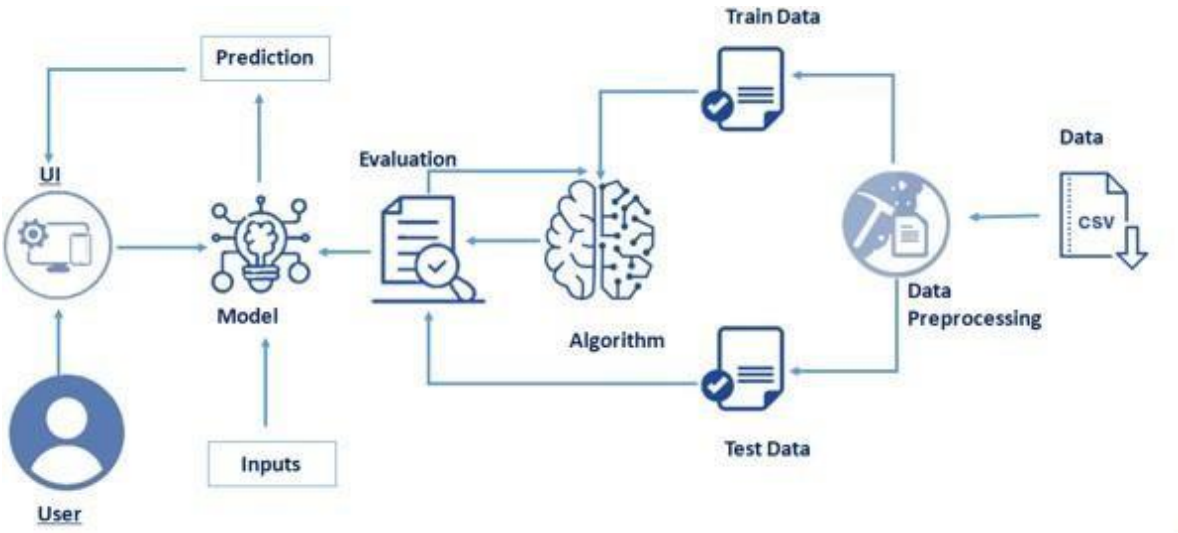
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly Interaction and affective UI No technical prerequisite from user
NFR-2	**Security**	Automatic logout when the app is not in use toprevent unauthorized access to the user. Users with valid login credentials will be able to access the site.
NFR-3	**Reliability**	The predictor system will be consistent in order for the system to produce trustworthy and accurate outcomes. Model accuracy is high
NFR-4	**Performance**	The performance will be more effective because logistic regression is applied .
NFR-5	**Availability**	Fast and efficient Minimal data redundancy
NFR-6	**Scalability**	Works good under multiple requests

5.PROJECT DESIGN

Data Flow Diagram



Solution & Technical Architecture



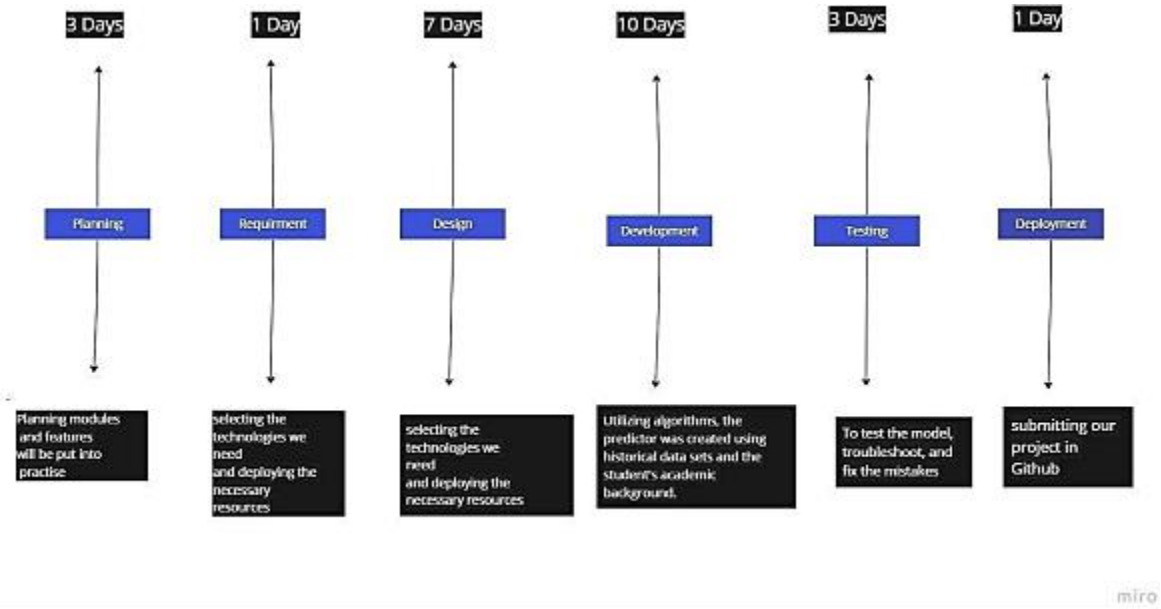
User Stories

User Stories :

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can view the cut off marks of previous years in my dashboard	I can access and download the files	High	Sprint-1
		USN-2	As a user, I can receive university details and their ranking.	I can only view(read-only)	Medium	Sprint-1
		USN-3	As a user, I can review the experience of the students in the university	I can access the review sections	Medium	Sprint-2
		USN-4	As a user, I can upload my documents.	I have read and write access to upload files.	High	Sprint-1
		USN-5	As a user,I can fill out the general and educational details in the form provided	I have read and write access to the forms filled.	High	Sprint-1
	Predictor	USN-6	I can view the list of universities in which I am eligible to get an admission	I can receive the final result as whether eligible or not	High	Sprint-2
		USN-7	I can view the list of universities I am eligible with the same cut-off but in previous years	I can access the files with read-only permission	Medium	Sprint-2

5.PROJECT PLANNING AND SCHEDULING

Sprint planning and estimation



Sprint delivery schedule

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	15	6 Days	24 Oct 2022	29 Oct 2022	15	29 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Home	USN-1	As a user, I will be able to view the details of the predictor.	8	Low	Aysha Nivera A, Amala Selciya T L
Sprint-1	Data Set	USN-2	Performing Data Analysis ofdataset and choosing a perfect model for prediction	12	High	Aysha Nivera A, Amala Selciya T L , Abina R
Sprint-2	Designing User Interface page	USN-3	As a user, we can enter the details of the mark topredict universities	13	Medium	Aysha Nivera A,
Sprint -3	Implementing ML model	USN-4	The user details will be validated depending upon theaccuracy and efficiency of the ML model	12	High	Aysha Nivera A, Aswathy P M
Sprint-3	Python With Flask	USN-5	For Backend and frondend, integrate ML model with Flask.	13	High	Aysha Nivera A, Amala Selciya T L
Sprint-4	Predicted result page	USN-6	As a user, I can get the prediction on the resulted page	14	Low	Aysha Nivera A, Abina R

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 per day)

Av=60 story points/24 days=2.5

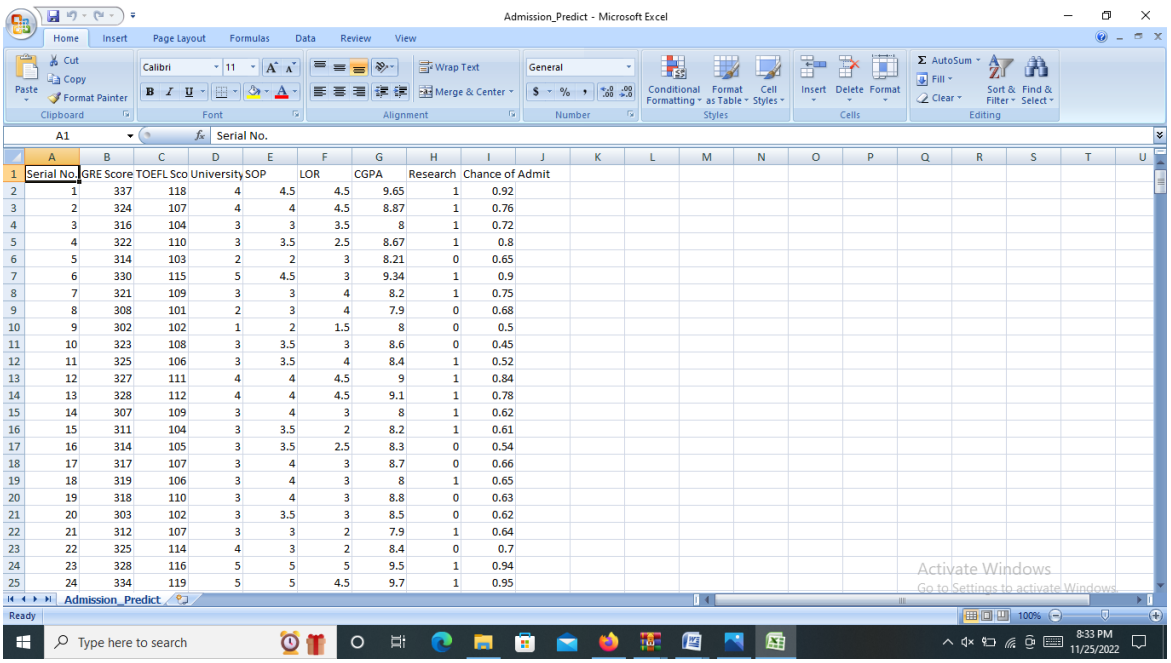
Burn down Chart:

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 can be applied to any project containing measurable progress over time.

4.CODING AND SOLUTIONING

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 Sco	University	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

8 TESTING

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

User Acceptance Testing

1. 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).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, andhow 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 RESULT ANALYSIS

Perfomance metrics

Importing libraries

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

Python

```python
data = pd.read_csv('C:\\Users\\shadow\\OneDrive\\Desktop\\ibm project\\Development P
```

Python

```python
data.head()
```

Python

	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

```python
data.info()
```

Python

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

Handling Missing Values

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

Python

```
Serial No.           0
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

Data Visualization

```python
data["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))

plt.title("GRE Scores")
plt.xlabel("GRE Score")
plt.ylabel("Frequency")

plt.show()
```

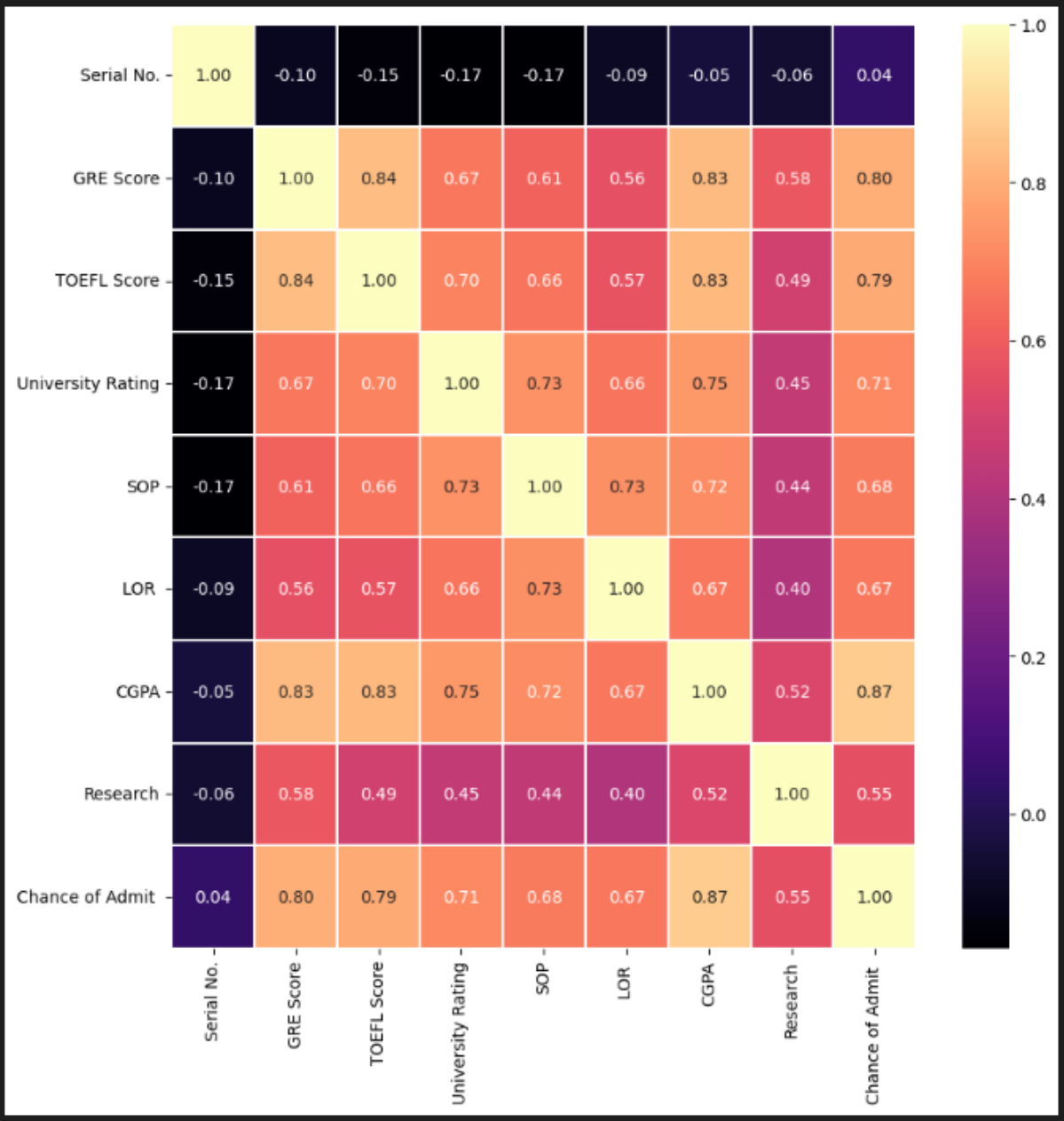
Python

```python
plt.figure(figsize=(10, 10))

sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")

plt.show()
```

Python



Training and Testing Split

Markdown

```python
X=data.drop(['Chance of Admit '],axis=1)
y=data['Chance of Admit ']
```

Python

```python
X
```

Python

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	1	337	118	4	4.5	4.5	9.65	1
1	2	324	107	4	4.0	4.5	8.87	1
2	3	316	104	3	3.0	3.5	8.00	1
3	4	322	110	3	3.5	2.5	8.67	1
4	5	314	103	2	2.0	3.0	8.21	0
...	...	...	...	...	...	...	...	...
395	396	324	110	3	3.5	3.5	9.04	1
396	397	325	107	3	3.0	3.5	9.11	1
397	398	330	116	4	5.0	4.5	9.45	1
398	399	312	103	3	3.5	4.0	8.78	0
399	400	333	117	4	5.0	4.0	9.66	1

400 rows × 8 columns

MODELING AND TRAINING

```python
from sklearn.ensemble import GradientBoostingRegressor
rgr = GradientBoostingRegressor()
rgr.fit(X_Train,y_Train)
```

Python

```
▾ GradientBoostingRegressor
GradientBoostingRegressor()
```

```python
rgr.score(X_Test,y_Test)
```

Python

```
0.8924247349697868
```

```python
y_predict=rgr.predict(X_Test)
```

Python

```python
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)))
```

Python

```
Mean Absolute Error: 0.036856448408973445
Mean Squared Error: 0.002192668178355762
Root Mean Squared Error: 0.04682593489035496
```

LogisticRegression

```python
from sklearn.linear_model._logistic import LogisticRegression

lore = LogisticRegression(random_state=0, max_iter=1000)

lr = lore.fit(X_Train, y_Train)
```

Python

```python
y_pred = lr.predict(X_Test)
```

Python

```python
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_m

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))
```

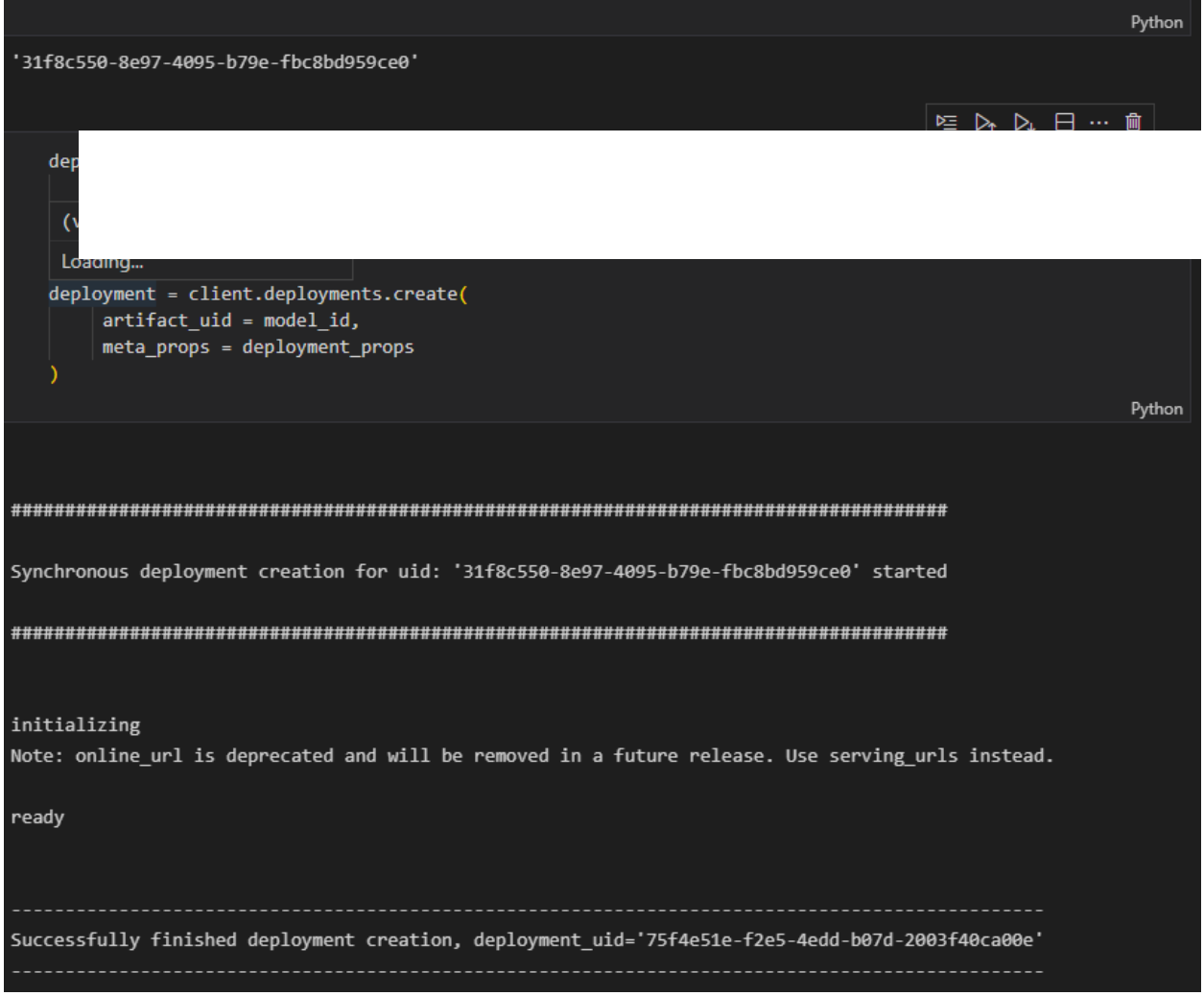
Python

```
Accuracy Score: 0.9333333333333333
Recall Score: 0.9259259259259259
ROC AUC Score: 0.962962962962963
Confussion Matrix:
 [[ 6  0]
 [ 4 50]]
```

```python
import joblib
joblib.dump(rgr,'model.pkl')
```

Python

```
['model.pkl']
```



SOURCE CODE:

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">
    <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\hat.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>
```

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
<strong>{{content[0]}}%</strong> chance</p>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>

{% endblock %}
```

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 only have
<strong>{{content[0]}}%</strong> chance</p>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>
```

```
{% endblock %}
```

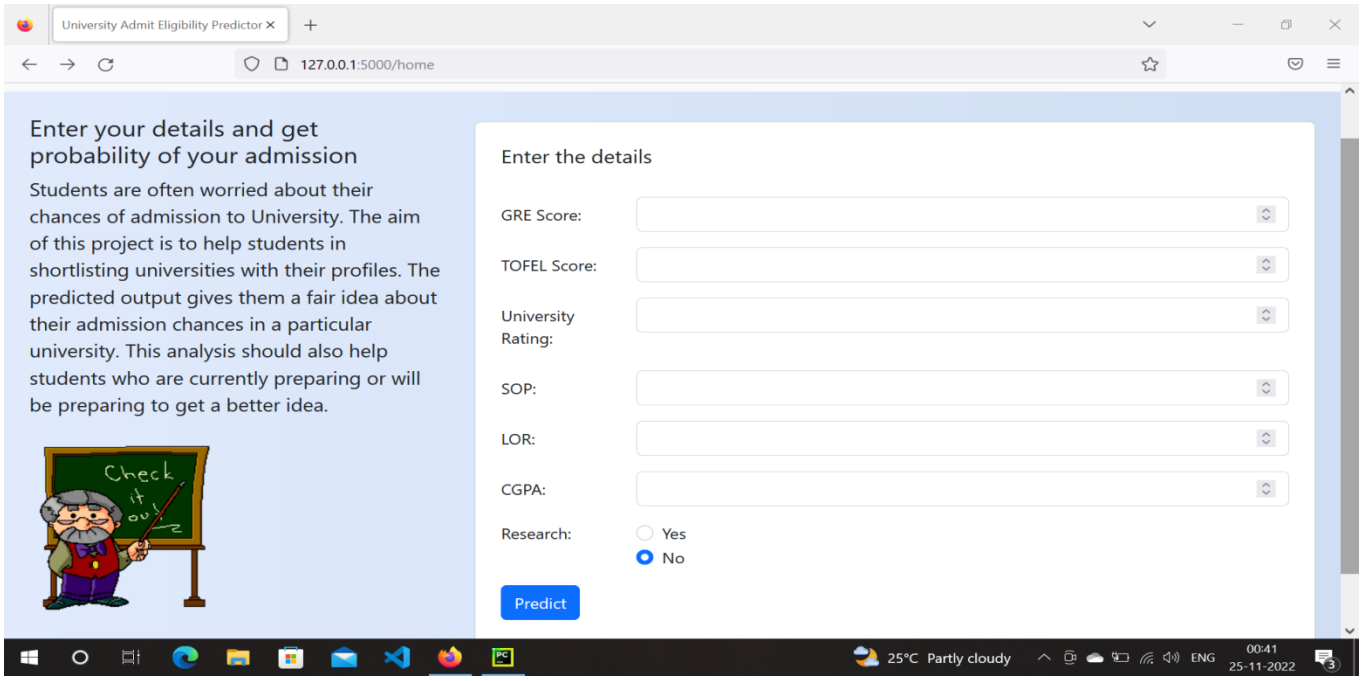
Demo2.html:

```
{% 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/animated-teach.gif" border="0" 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" placeholder="250 to 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" placeholder="50 to 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" placeholder="1
to 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" placeholder="1 to 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" placeholder="1 to 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" placeholder="5 to 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>
```

```
<script type="text/javascript" src="../static/js/script.js" async></script>
{% endblock %}
```

OUTPUT IMAGES



University Admit Eligibility Pred

127.0.0.1:5000/home

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.

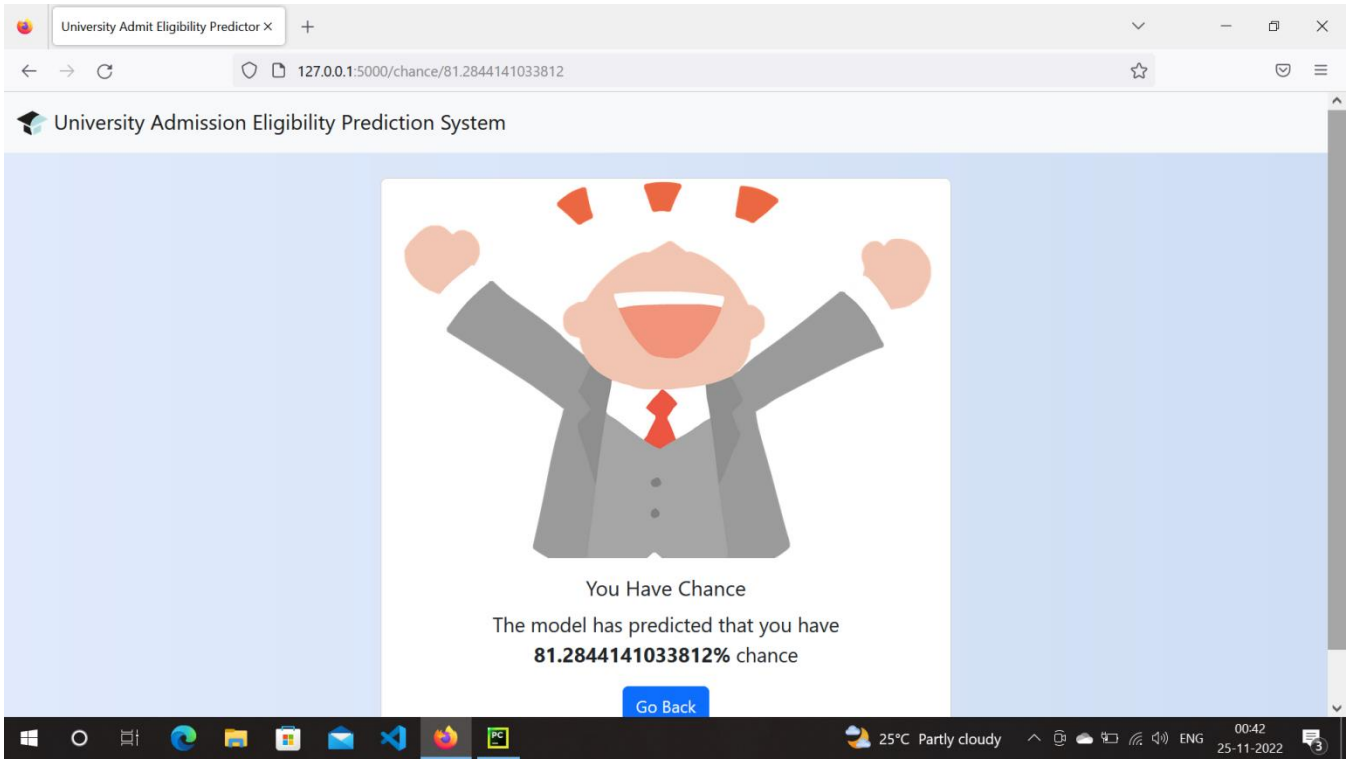
Enter the details

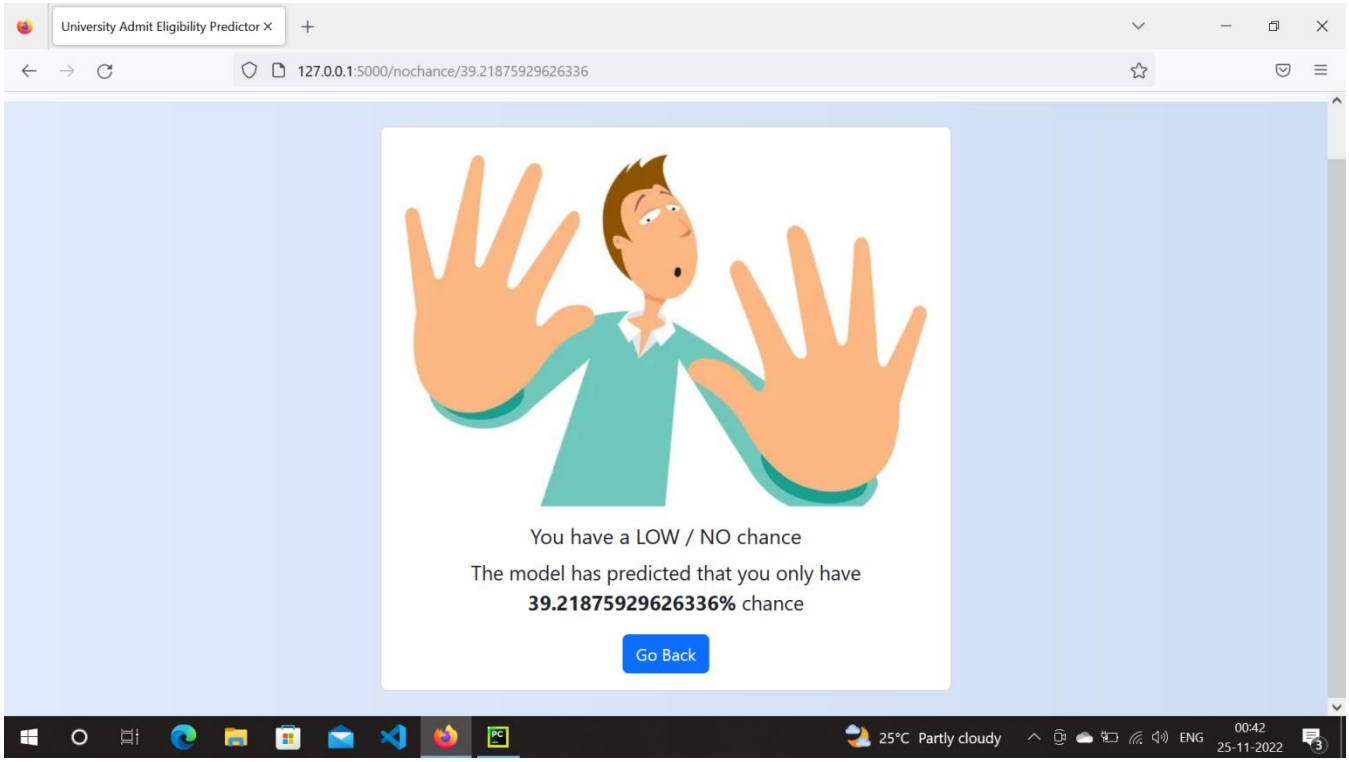
GRE Score:	300
TOFEL Score:	65
University Rating:	2
SOP:	2
LOR:	2
CGPA:	6
Research:	○ Yes ◉ No

Predicting...

127.0.0.1

25°C Partly cloudy ENG 00:42 25-11-2022


University Admit Eligibility Predictor
127.0.0.1:5000/chance/81.2844141033812
University Admission Eligibility Prediction System
You Have Chance
The model has predicted that you have
81.2844141033812% chance
Go Back
25°C Partly cloudy
ENG
00:42
25-11-2022


University Admit Eligibility Predictor
127.0.0.1:5000/nochance/39.21875929626336
You have a LOW / NO chance
The model has predicted that you only have
39.21875929626336% chance
Go Back
25°C Partly cloudy
ENG
00:42
25-11-2022

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 = "wf8mge_OQdwVO8ao2kmWCtfxOfLWl8442SH44V85v2Ls"
        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/8308fd4c-24a5-46ab-96fa-
263657ae4ad0/predictions?version=2022-10-18',
            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])
```

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

if __name__ == "__main__":
    app.run()
```

Java Script:

```javascript
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;
```

10. ADVANTAGES & DISADVANTAGES

a. Advantages

1. It helps student for making decision for choosing a right college.
2. Here the chance of occurrence of error is less when compared with the existing system.
3. It is fast, efficient and reliable.
4. Avoids data redundancy and inconsistency.
5. Very user-friendly.
6. Easy accessibility of data.

b.Dis-Advantages

1. Required active internet connection.
2. 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 endlanguages 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 wereadded, 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 improvementas well as remembering everything that had to be done was a project in itself. Debugging cansometimes be a relatively straight forward process, or rather rather finding out what you mustdebug can be. Since so many parts of the admissions system are integrated into one another, if anerror 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 downthe processand can be frustrating if the apparent cause of a problem is not obvious at first. Language usedmust 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 ofthe 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. The subject of this examination was to determine if the below variables contribute to the admission of student to Master's degree program.

GRE Score
TOEFL Score
University Rating
SOP

LOR
CGPA

The results of this examination appear to indicate that it greatly contributes to the response variable 'Chance of Admit'. Higher the **GRE, TOEFL** score then higher the admit chances. The model predicts 87.5% accuracy and can be used for predicting the admit chances based on the above factors. This model will be helpful for the universitiesto predict the admission and ease their process of selection and timelines.

As part of the hypothesis, the model proved that admission to Master's degree program isdependent on **GRE, TOEFL** and other scores. This model would likely be greatly improved by the gathering of additional data of students from different universities which has similar selection criteria to choose the candidates for Master's program.

12.FUTURE SCOPE

The future scope of this project is very broad. Few of them are:

1. This can be accessed anytime anywhere, since it is a web application provided only aninternet connection.
2. 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

University Admission depends on many factors, among them GPA, GRE and TOEFL are most important. We have used these three features in this project but more features can be implemented to get more accurate result. Other features could be Statement of Purpose (SOP), Letter of Recommendation (LOR), industry experience, internship experience, papers published, journals published etc. Also, as an extension to this work, recommendation of university with respect to research interest can be made with further study. As mentioned before the data we actually fed to model is far less than the total data set as we first filter it based on user's intended major . So, we are working on insufficient dataset. Even though the accuracy is ranged between 70 % - 85%. the result will never be satisfied to the user. So, we can work on huge data set for that we can scrape the data from thegradcafe.com and edulix.com. We can also add more features to our system like University ranking, University News feed. This project is focused for the students intending to pursue master and PhD degrees, we can use dataset of undergraduate college admission and use it for the students intending to pursue bachelor degree.

REFERENCE:

Installation of Anaconda Navigator:https://www.youtube.com/embed/5mDYijMfSzs

Installation Of Pycharm Professionals*:https://www.youtube.com/embed/z73PyNDgVyQ*

Installation Of Python Packages: https://www.youtube.com/embed/akj3_wTploU

Data Collection: https://www.kaggle.com/datasets/rishal005/admission-predict

Data Pre-processing:https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf
Handling Null Values:https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e

Logistic regression : https://www.youtube.com/watch?v=yIYKR4sgzI8&t=76s

Register in IBM cloud: https://youtu.be/x6i43M7BAqE

Train Machine Learning Model on IBM Watson: https://youtu.be/TysuP3KgSzc

Integrate Flask With Scoring Endpoint: https://youtu.be/ST1ZYLmYw2U