UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

A PROJECT REPORT

TEAM ID: PNT2022TMID39283

Submitted by

Date	19 November 2022
Team ID	PNT2022TMID39283
Team Leader	Devadharshini M (422619104007)
Team Members	Kirubaharan R (422619104024) Hariprasath S (422619104014) Kavithas R (422619104021) Kalaidharani V (422619104020)
Project Name	UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

CONTENTS

1. Introduction	
	1.1 Project Overview
	1.2 Purpose
2. Literature Survey	
	2.1 Existing Problem
	2.2 References
	2.3 Problem Statement Definitions
3. Ideation & Proposed Solution	
	3.1 Empathy Map Canvas
	3.2 Ideation & Brainstorming
	3.3 Proposed Solution
	3.4 Problem Solution Fit
4. Requirement Analysis	
	4.1 Functional Requirement
	4.2 Non-Functional Requirement
5. Project Design	
	5.1 Data Flow Diagrams
	5.2 Solution & Technical Architecture
	5.3 User Stories
6. Project Planning & Scheduling	
	6.1 Sprint Planning & Estimation
	6.2 Sprint Delivery Schedule
	6.3 Reports From JIRA
7. Coding & Solutioning	
	7.1 Feature 1
	7.2 Feature 2
8. Testing	
	8.1 Test Cases
	8.2 User Acceptance Testing
9. Results	
	9.1 Performance Metrics

11. Conclusion		
12. Future Scope		
13. Appendix		
••	13.1 Source Code	
	13.2 GitHub	
	13.3 Project Demo Link	

1. INTRODUCTION

The world's business sector is escalating and is constantly seeking information and experiences that are commonly beneficial to individuals. Young specialists who need to stay in their current positions are alwayslooking for advanced degrees to help them address their skills and information. As such, the number of hersophomores 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 Researchscholarships 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 andmaster's degrees in computer science at US universities. The focus of this study applies to theseundergraduate 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 340foci.
- English as a Foreign Language (TOEFL) test score. It consists of 120priority areas.
- 3. Uni.Rating. Shows the position of colleges offering bachelor's degrees among various colleges. Your scorewill 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.
- 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.
- 7. 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.

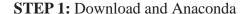
1. PROJECT OVERVIEW

1.1 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.

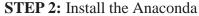
WAY TO INSTALL ANACONDA:

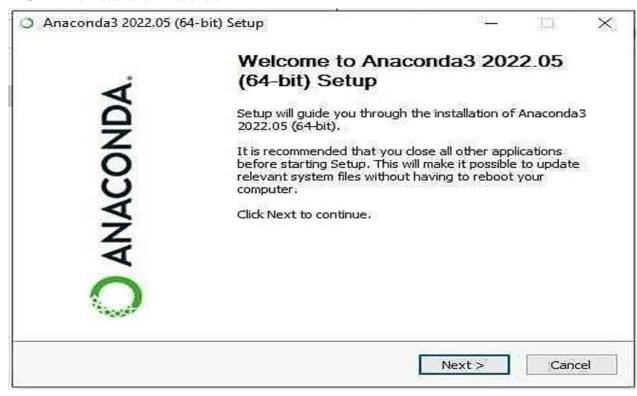






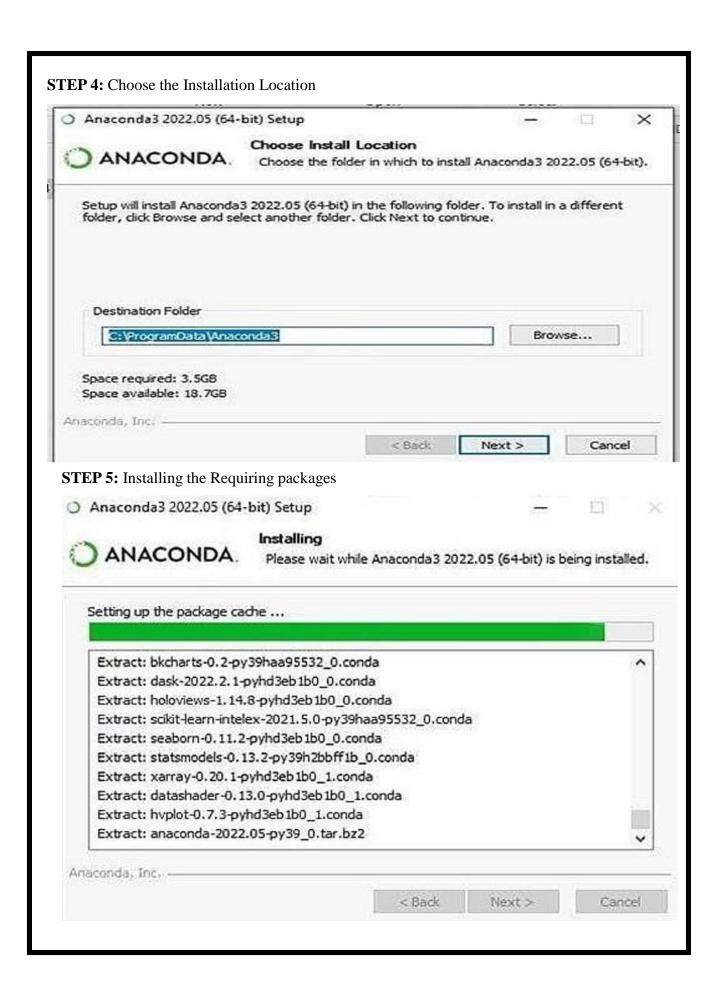
Contact Sales



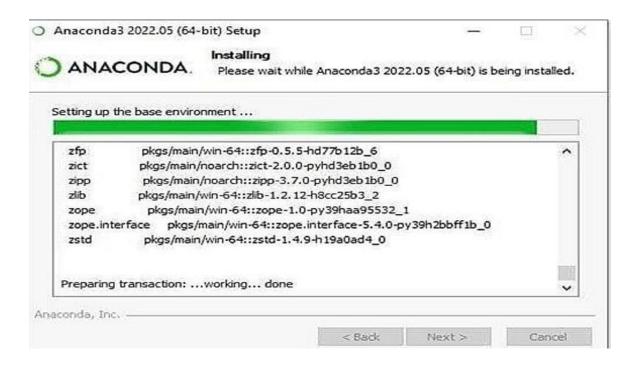


STEP 3: Click I Agree

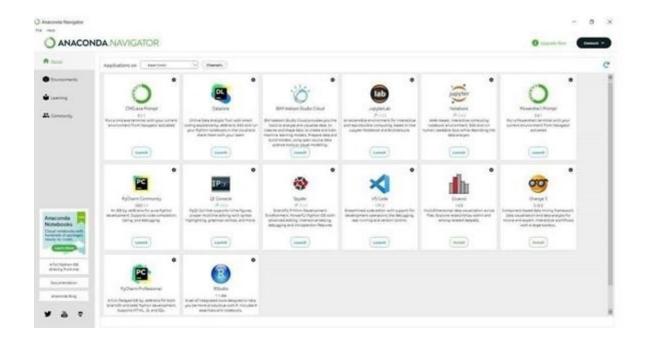




STEP 6: Setting up the base environment

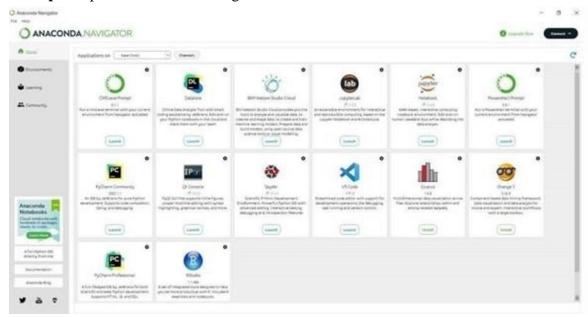


STEP 7: Successfully Installed and check the Anaconda Navigator working or not

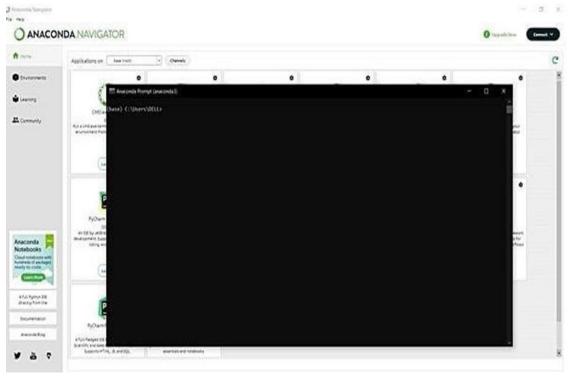


Python packages installation:

Step 1: Open the anaconda navigator in the start menu



Step 2: Open the CMD.exe prompt



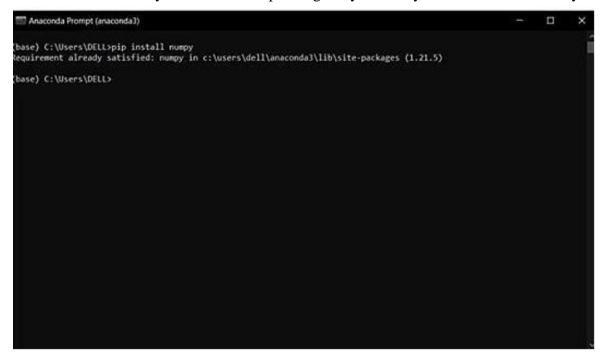
Step 3: Install the NUMPY package

To enter the numpy package enter the command in

the CMD.exeCommand: **Pip install numpy**

Numpy:

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



Step 4: Install the pandas package.

To enter the pandas package enter the command in

the CMD.exeCommand: **Pip install pandas**

Pandas:

Pandas is one of the most widely used Python libraries for data science. It provides powerful and easy-to-use structure and data analysis tools. This package comes pre-installed with Anaconda. An open source library built on top of the NumPy library. A Python package that provides various data structures and operations for

working with numerical data and time series. Mainly, it's common for data to be imported and analyzed much easier. Pandas is fast, providing users with high performance and productivity.

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

(base) C:\Users\DELL>pip install pandas

Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.4.2)

Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.2)

Requirement already satisfied: numpy>=1.18.5 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (1.21.5)

Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2021.3)

Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->panda
s) (1.16.0)
```

Step 5: Install the Matplotlib package.

To enter the Matplotlib package enter the command In

the CMD.exeCommand: 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.

Step 6: Install the Scikit-learn package.

To enter the Scikit-learn package enter the command in

the CMD.exeCommand:**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.exeCommand: 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.

```
Anaconda Prompt (anaconda3)

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

PROJECT FLOW

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

- 1. User interacts with the UI (User Interface) to enter Data
- 2. The entered data is analyzed by the model which is integrated
- 3. 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

- 4. Data Collection.
 - a. Collect the dataset or Create the dataset
- 5. Data Preprocessing.
 - a. Import the Libraries.
 - b. Importing the dataset.
 - c. Checking for Null Values.
 - d. Data Visualization.
 - e. Taking care of Missing Data.
 - f. Label encoding.
 - g. One Hot Encoding.
 - h. Feature Scaling.
 - i. Splitting Data into Train and Test.
- 6. Model Building

- a. Training and testing the model
- b. Evaluation of Model
- 7. Application Building
 - a. Create an HTML file
 - b. Build a Python Code

PROJECT OBJECTIVES

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

PROJECT STRUCTURE

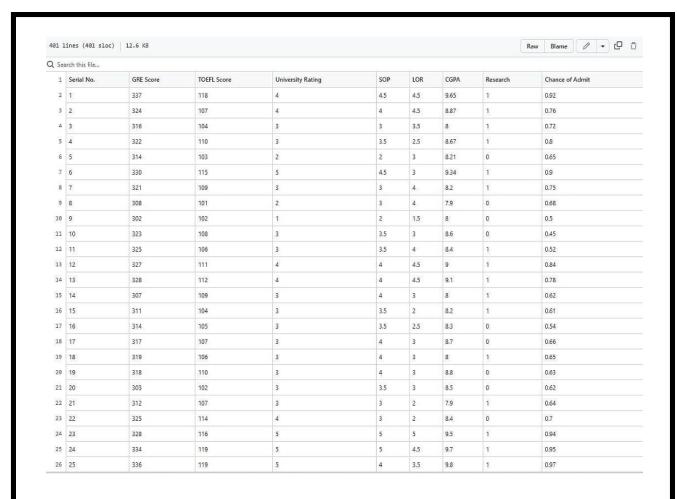
Create a Project folder that contains files as shown below

dataset	09-09-2020 23:56
Admission Predicties	01-08-2019 10:45
output -	10-09-2020 13:46
— E 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 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 themost commonly used repositories for sorting

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



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 interfacefor drawing attractive and informative statistical graphics.

Matplotlib-

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

Reading the Datasets:

You mighthave your data in .csv files, .excelfiles

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 stringto be interpreted as actualbackslashes 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 ofthe hidden information such as null values in a column or a row Check for the nullvalues. if it is present then the following steps can be performed

- Imputing data using the Imputationmethod 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
 LOR
                     False
                     False
 CGPA
 Research
                     False
 Chance of Admit
                     False
 dtype: bool
```

- If the dataset contains null values then the above functionsreturn as true. But if you lookat the dataset you can observe that the dataset does not have any null values.
- You can also check the number of null values presentin 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 importantas 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 librariescalled 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 uselloc of pandas (used to fix the indexes for selection) which takes two parameters —
[rowselection, column selection].

Let's split our datasetinto 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



 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, firstsplit the modelinto 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 datasetthat 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 theother 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 machinelearning algorithm.

- Train Dataset:Used to fit the machine learningmodel.
- Test Dataset:Used to evaluate the fit machinelearning 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 afew 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 instanceused 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 academictranscripts 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 nonfunctional 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 coupleof friends who are in panic mode waiting for a callfrom the universities they've appliedat.

This made me think— How can we predictwhether a studentwill get an admitor not? What arethe 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 recommenduniversities 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 togetherwith 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.

►<u>https://dl.acm.org/doi/10.1145/3388818.3393716</u>

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 postgraduateadmission 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 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 systemswhich handle the admission processaccurately from both perspectives.

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

Abstract:

- i. Analyzed university admissionstatistics.
- ii. Developed tools for matchinguniversity (in percentile)using CGPA, GRE (Verbal, Quantitative, AnalyticalWriting)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.

b.Problem Statement Definition

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

2.IDEATION AND PROPOSEDSOLUTION

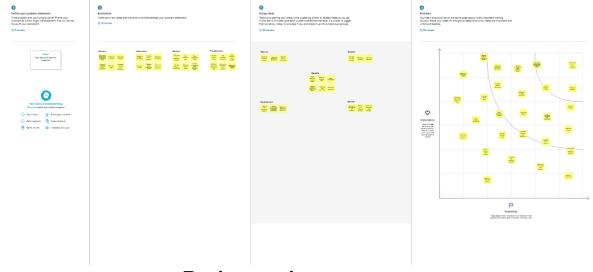
Ideation is the process where you generate deas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process. In this project the ideation phase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

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

3.1 Empathy Map



3.2 Ideation & Brainstorming



Brainstorming

2.1 Proposed Solution

			Students frequently have a number of inquiries when making
		Problem	educational plans regarding the programmes, colleges, employment
	Statement		prospects, costs involved, etc. One of their biggest concerns is getting
	1.	(Problem to be	into the university of their dreams. Students frequently choose to
		solved)	further their study at institutions with a solid international reputation.
			Concerns about getting into college are common among students.
•			The main goal of this research is to create a mechanism to address the
			issues students encounter when applying to institutions. We are
		Idea / Solution	creating a system called the University Admit Eligibility Predictor
	2.		(UAEP) that will enable students to estimate, based on their profiles,
		description	the likelihood that a certain university would select their application.
			The system will also suggest universities for the student to apply to that
			have a good chance of admitting him or her.
			We'll be developing a straightforward user interface that will enable
			users to enter information about a student's profile and receive the
	3.	Novelty /	application's forecasted result as an output. This project's goal is to
	ა.	Uniqueness	assist students in narrowing down institutions based on their profiles.
			The anticipated results offer them a good indication of their prospects
			of admission to a particular university.
		I	

		Students will eventually benefit from this research by not having to spend
	Social Impact	as much time and money at education consulting firms. Additionally, by
4	and Customer	suggesting institutions where they have the best chance of being
4.	Satisfaction	admitted, it will help them reduce the number of applications they submit,
		saving them money on application fees.
	Business	
_	Model	This programme allows for the placement of advertisements for various
5.	(Revenue	universities, which allows us to make money.
	Model)	
		Future plans include the addition of one faculty person from each
	Scalability of	institution, who will assist students in getting information about the school,
6.	the Solution	the courses, and other topics. In this software, a chat feature is also an
		option.

2.1 Problem Solution Fit

Problem-Solution fit University Admission Predictor AS 1. CUSTOMER SEGMENT(S) 5. AVAILABLE SOLUTIONS 6. CUSTOMER CONSTRAINTS · Customersegment are Student Community ebsitesSolutionare: Unclassified data Cut-off mark criteria unknown Universities Cut-off mark criteria unknown University Requirements unknown Parents Constaints: Cost of living, tuition fee budget Details about the locality University Constaints: Why students do & do not prefer What are the factors that hold back students from applying What are required to enable students apply to universities No proper tier/ ranking is given Inaccurate/ Misleading information on websites Limited number of searches Counselling Services Solution are: Define CS, Expensive Might not get into preferred universities 9. PROBLEM ROOT CAUSE 7. BEHAVIOUR 2. JOBS-TO-BE-DONE / PROBLEMS Look out for websites with accurate and elaborate details Compare various details and cross verify Give several data as input and improve their search results Ask queries and look out for immediate response Privacy and Security of the website is ensured Job to be done are Data acquisition Accuracy Maintenance Security and Privacy handling Verification of acquired details Root cause problem students or parent and list the universities available for their cut-off mark so that the student or parent will be directly benefitted by preparing themselves in par with the university's requirement. Query handling СН TR 10. YOUR SOLUTION SL 3. TRIGGERS 8. CHANNELS of BEHAVIOUR 8.1 ONLINE Review and Feedback Choice of preference A website is created where the users can login and give the details as input. Acceptance rate Feedback and reviews details as input. Based on the data collected, universities above the ranking, universities on their range and below the range are displayed. Apart from that, data such as scholarships, accommodations, and cost of living are also displayed. Universities can upload what are their requirement from students. Preferred suggestions 4. EMOTIONS: BEFORE / AFTER EM 8.2 OFFLINE Real-time user experience Mock tests scores are uploaded and based on which the A checkbox to add a university to preference and a drop box to select the why they prefer. After Confident Clear about career prospect

3.REQUIREMENT ANALYSIS

3.1 Functional Requirements

Following are the functional requirements of the proposed solution.

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

3.2 Non-functional Requirements:

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

FR No.	Non-Functional	Description
	Requirement	
NFR-1	Usability	a. A logical interface is essential to
		makeeasyuse ofsystem,
		speeding up common tasks.
		b. The product could be used by two
		categories of

		people mainly administrator category
		andotherusers.
NFR-2	Security	Some of the factors that are identified
		toprotectthe software from accidental or
		malicious access,use, modification,
		destruction, or disclosure are described
		below:
		Keep specific log or history datasets.
		 b. Utilize certain cryptographic techniques.
		c. Restrict the no of systems that
		can accessthe online admission
		system site. This could be done
		only by registering the systems
		physical
		addresses

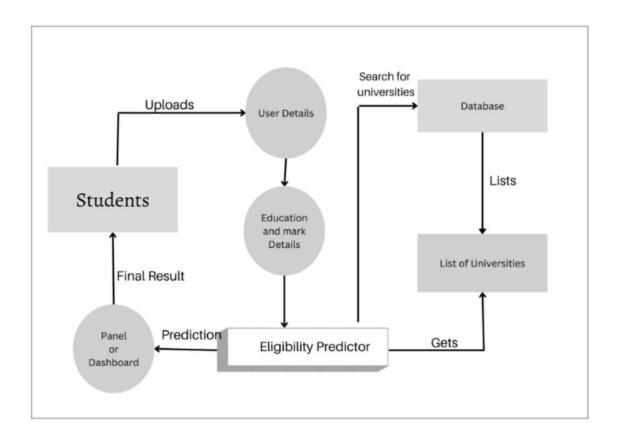
		before using them for online				
		admissionprocess.				
		a. Check data integrity for				
		criticalvariables.				
		b. Every user should be licensed				
		to usethe systemunder any of				
		the four categories provided i.e.				
		either verifier or advisor or local				
		counsellor or administrator.				
		c. Communication needs to be				
		restrictedwhen the				
		application is validating the user or				
		license.				
NFR-3	Reliability	a. All data storage for user				
		variables willbecommitted to				
		the database atthe timeof				
		entry.				
		b. Data corruption is				
		prevented byapplying				
		thepossible backup				
		procedures and techniques.				

NFR-4	Performance	a. The database should be
		able toaccommodate
		aminimum of 10,000
		records of students.
		b. At any instant the system should
		supportuse ofmultiple usersat a
		time.
		c. Availability results of the
		requestedcollege should be
		presented to the student in
		max of twoseconds, so
		retrieving of data should
		bereliable.
		d. As each student will be givena maximum timeof10min,
		accessing fromthe
		database
	<u> </u>	should be
		done at relevant
		speed.

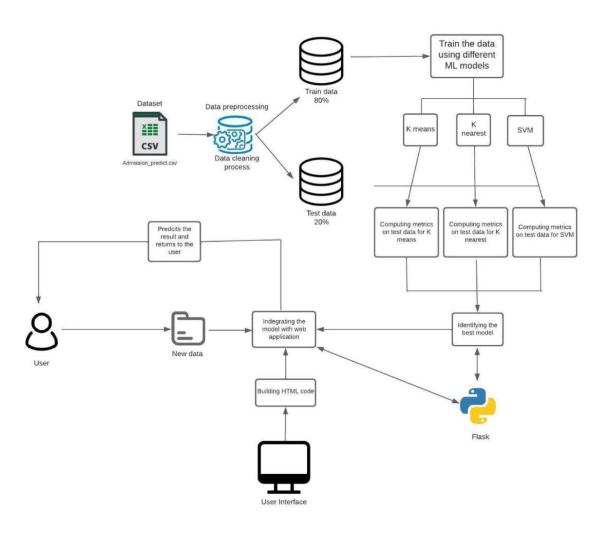
	Availability	The system should available at all the
NFR-5		timemeaning that the user can access
		easily.
		Increase of the hardware and data base
		failurea replacement page will be show and
		for database back should be
		retrieved fromdata folder.
NFR-6	Scalability	Assesses the highest workloads
		underwhich the system will still meet
		theperformance Deals withthe measure of
		the system's response timeunder different
		loadconditions requirements.
		Example:
		The system must be scalable enough
		tosupport 1,000,000 visitsat the same
		timewhilemaintaining
		optimal performance.

4. PROJECT DESIGN

4.1 Data Flow Diagram



4.2 Solution & Technical Architecture



4.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Student)	Dashboard	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 view university details and their rankings	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-2
	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
Administrator	Dashboard	USN-8	As an administrator, I can have access to update the latest updates of the universities	I can have access to read and write the university information in the dashboard	High	Sprint-3
		USN-9	As an administrator, I can access any resources available in the page	I can access the resources that are available	Medium	Sprint-3
		USN-10	As an administrator, I can have a track on the universities the student is eligible to get admission at	I can access the list of the universities obtained as final result	High	Sprint-3

5. PROJECT PLANNING & SCHEDULING

5.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint-1	Pre requisites	USN-1	Install Anaconda, install python packages, prior knowledge	4	Low	Kirubakaran. R
Sprint-1	Pre requisites	USN-2	Project flow, project objectives, project structure	4	Low	Kirubakaran. R
Sprint-1	Data collection	USN-3	Download the dataset	4	Medium	Kirubakaran. R
Sprint-2	Data collection	USN-4	Importing TheLibraries, Reading The Dataset Analyze The Data, Handling Missing Values	4	Medium	Devadharshini. M
Sprint-2	Data collection	USN-5	Data Visualization	4	High	Devadharshini. M
Sprint-2	Data collection	USN-6	Splitting Dependent And Independent Columns, Splitting the Data into train and test	5	Medium	Devadharshini. M

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team Members
Sprint-2	Model Building	USN-7	Training and Testing The Model	5	Medium	Hariprasath. S
Sprint-2	Model Building	USN-8	Model Evaluation	5	High	Hariprasath. S
Sprint-2	Model Building	USN-9	Save the model	5	Low	Hariprasath. S
Sprint-3	Application Building	USN-10	Build HTML Code	10	High	Kavithas. R
Sprint-3	Application Building	USN-11	Build HTML Code	10	High	Kavithas. R
Sprint-4	Train the Model on IBM	USN-12	Register for IBM Cloud	4	Medium	Kavithas. R
Sprint-4	Train the Model on IBM	USN-13	Train Machine Learning Model on IBM Watson	8	High	Kalaidharani. v
Sprint-4	Train the Model on IBM	USN-14	Integrate Flask with scoring endpoint	8	High	Kalaidharani. v

5.2 Sprint Delivery Schedule

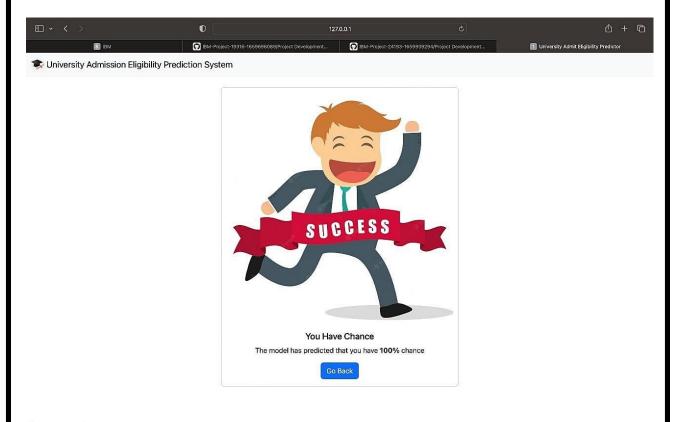
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

5.3 Report from JIRA > How to read this report Sprint burndown chart Estimation field ✓ Story points ✓ ABDFBI Sprint 1 Date - October 24th, 2022 - October 29th, 2022 Guideline Ideal burn rate ____ Remaining work 15 -Oct 25 Oct 26 Oct 27 Oct 28 Oct 29 Date

6. CODING & SOLUTIONING

6.1 Feature 1

The new featurewill predict the chances in the admission of the university. Thefeature was designed in thehtml code connected with app.py as the backend.



Source Code:

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

{% block body %}

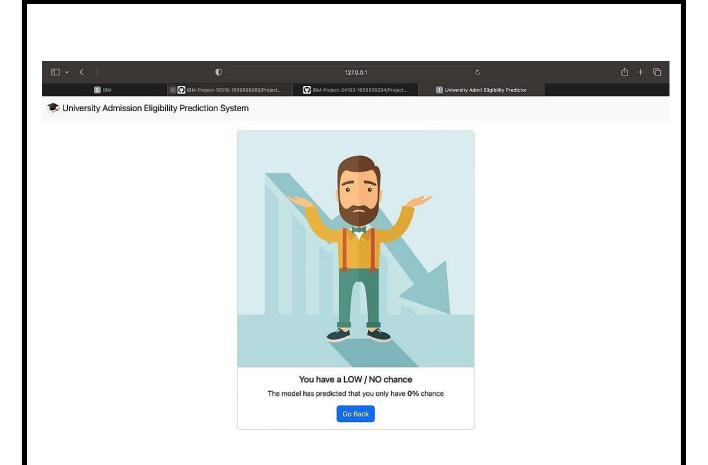
<div class="container text-center p-4">

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

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

6.2 Feature 2

The new featurewill predict the low chancesin the admission of the university. Thefeature was designed in the html code connected with app.py as the backend.



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;">

<ing 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>
```

```
The modelhas predicted that you have no chance
<a href="/home" class="btn btn-primary">Go Back</a>
</div>
</div>
</div>
{/div>
{/div>
</div>
</div>
</ri>
{% endblock%}
```

6.3 Database Schema

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

arch this file								
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

7.TESTING

7.1 Test Cases

Test Case Analysis

This report showsthe 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 ReportOutput	4	0	0	4
Version Control	2	0	0	2

7.2 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, 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 Reproduc ed	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 showsthe 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 ReportOutput	4	0	0	4
Version Control	2	0	0	2

8.RESULTS

8.1Performance Metrics

Measure the performance using Metrics

```
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

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
```

9.ADVANTAGES & DISADVANTAGES

9.1 Advantages

- 1. It helps student for making decision for choosing aright 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.

9.2 Dis-Advantages

- 1. Required active internet connection.
- 2. System will provide inaccurate results if data enteredincorrectly.

10. 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 methodsto improve the usability of the systemmade themselves apparent. Furthermore, adding one feature meant that another required feature wasnow 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 ratherfinding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if anerroroccurs on one page, it may be a displayerror, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in thedatabase initially, and all three must be checked on each occasion. This slows down the processand can be frustrating if the apparent cause of a problem is not obvious at first. Languageusedmust be simple and easy to understand and compatibility is paramount. If this system were notdesigned 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 featuresthat are includedwork flawlessly and the errors that do exist are minor or graphical.

11. FUTURE SCOPE

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

- 1. This can be accessed anytimeanywhere, since it is a web application provided only aninternet connection.
- 2. The user had not need to travela long distance for the admission and his/her time is also saved as a result of this automated system.

12. APPENDIX

12.1 Source Code

PYTHON CODE

Uploading the pythoncode,

IMPORT STATEMENTS

In [1]:

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

LOAD THE DATASET

body)

```
In [2]:
import os, types
import pandas as pd
from botocore.client
import Configimport
ibm_boto3
  def__iter_
  (self): return 0
  #@hidden cll
 # The followingcode accesses a file in your IBM Cloud ObjectStorage. It includesyour
credentials.
  # You might want to remove those credentials beforeyou 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 acceptsbody as file-like object
  if not hasattr(body, " iter "): body. iter = types.MethodType( iter, body)
  data =
  pd.read_csv(
```

data.head()

Out[2]:

	Serial No.		TOEFL Score	University Rating	SOP	LOR	CGPA	Kesearch	Chance ofAdmit
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		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	Universit y Rating	SOP	LOR	CGPA	Research	Chance of
								Admit
count	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
	00	00	00	00	00	00	00	00
mean	316.8075 00	107.4100 00	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.47364 6	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.0000 00	92.00000 0	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.0000 00	103.0000 00	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.0000 00	107.0000 00	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.0000 00	112.0000 00	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.0000 00	120.0000 00	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [5]:

data.info()

Out[5]:

<class

'pandas.core.frame.DataFr ame'>RangeIndex: 400

entries, 0 to 399

```
Data columns(total 8 columns):
  # Column
                   Non-Null Count Dtype
   1. GRE Score
                      400 non-null int64
   2. TOEFL Score
                       400 non-null int64
   3. University Rating 400 non-null int 64
   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 float64dtypes:
      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

0

CGPA

Research

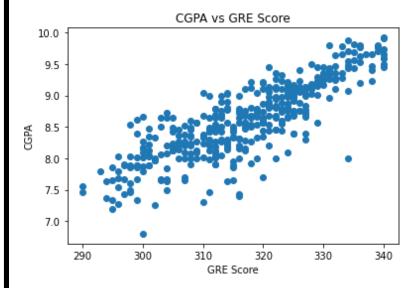
Chance of Admit 0 dtype: int64

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()

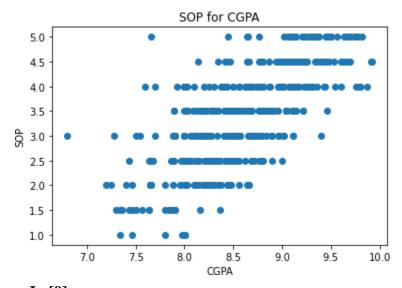
Out [7]:



In [8]:

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

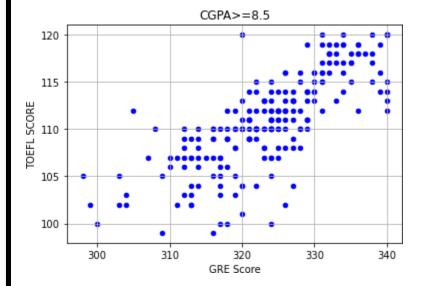
Out [9]:



In [9]:

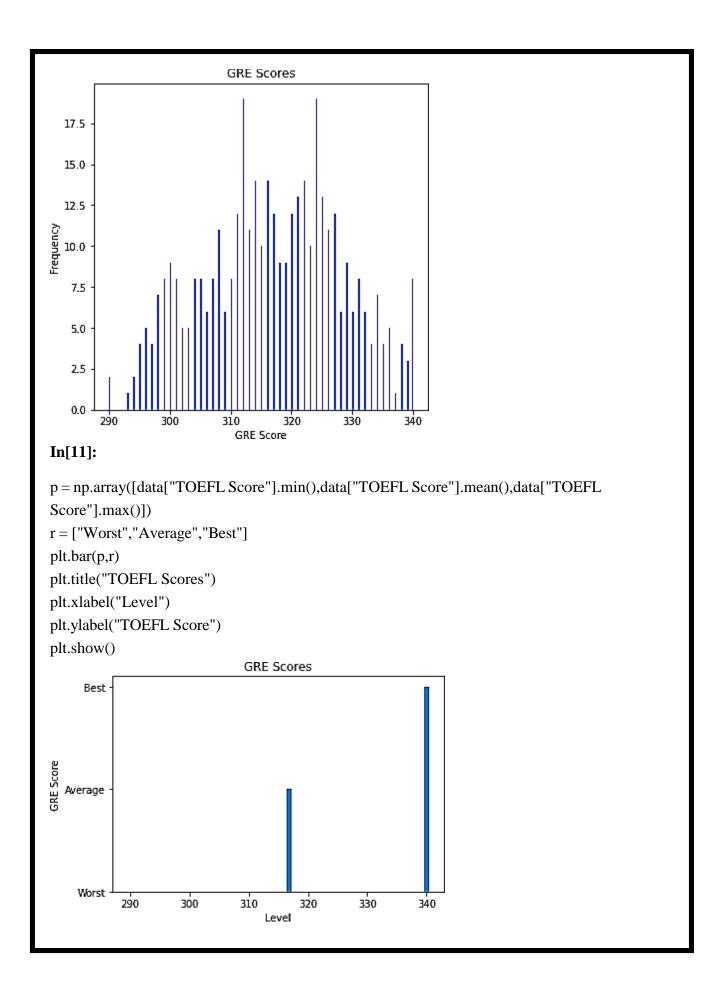
```
data[data.CGPA >= 8.5].plot(kind='scatter', x='GRE Score',
y='TOEFLScore',color="BLUE")
plt.xlabel("GREScore")
plt.ylabel("TOEFL SCORE")
plt.title("CGPA>=8.5")
plt.grid(True)
plt.show()
```





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()
```



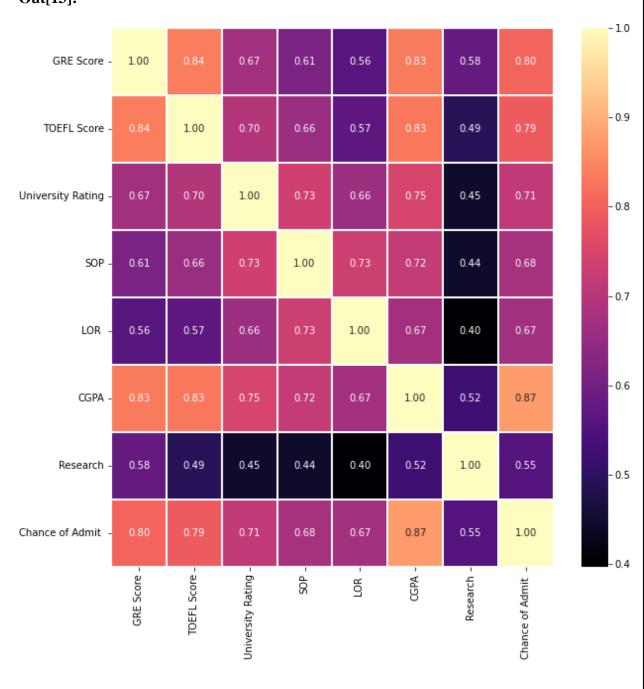
In[13]:

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

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

plt.show()

Out[13]:

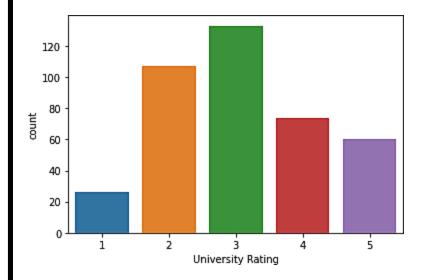


In[14]:

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

Out[14]:

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

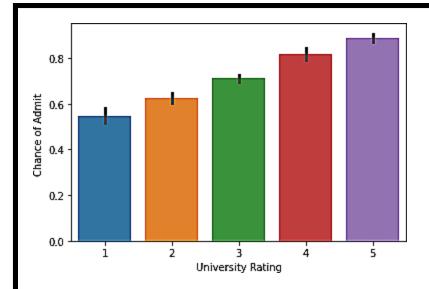


In[15]:

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_sety=data['Chance of Admit '] #output labels

In [17]:

from sklearn.model_selection importtrain_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

GradientBoostingRegressorrgr =

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 loadingthis 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 SquaredError:', np.sqrt(mean_squared_error(y_test,
  y_predict)))Mean AbsoluteError: 0.061115035673946834
  Mean Squared Error:
```

```
0.007194293635482686 Root Mean
  SquaredError: 0.08481918200196631
In [22]:
  y_train =
  (y_train>0
  .5)y_{test} =
  (y_{test}>0.
  5)
In [23]:
  from sklearn.linear_model._logistic
  importLogisticRegressionlore =
  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_matrixprint('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: 0.916666666666666 Recall Score: 1.0 ROC AUC Score: 0.72222222222222 Confus sion Matri x:[[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 alreadysatisfied: 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-

 $3.9/lib/python 3.9/site-packages (from ibm-watson-machine-learning) \ (1.26.7)$

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==2.11.0in /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 alreadysatisfied: 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 1.0.255Uninstalling ibm-watson-machine-
    learning-1.0.255:
     Successfully uninstalled ibm-watson-machine-
  learning-1.0.255Successfully installedibm-watson-
  machine-learning-1.0.256
  Note: you may need to restart the kernel to use updatedpackages.
In [31]:
  from ibm_watson_machine_learning
  import APIClientimport json
In [32]:
uml_credentials = {
  "url": "https://us-south.ml.cloud.ibm.com",
  "apikey": "Ct8qx91QEDmW9Xw2viaF7uGxI-jHrk0ogjK117b_L4-K"
}
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['meta
      data']['id'])
    return idr
In [34]:
  space_uid = guid_from_space_name(client,
  "uni admit")print(space uid[0])
424b35d3-9ffd-4890-a149-d7642a3f640c
In [35]:
  client.set.default_space(space_uid[0])
Out[35]:
  'SUCCESS'
In [36]:
  client.software_specifications.list()
  NAME
                                                     TYPE
                         ASSET ID
  default_py3.6
                         0062b8c9-8b7d-44a0-a9b9-46c416adcbd9
  base kernel-spark3.2-scala2.12
  020d69ce-7ac1-5e68-ac1a-31189867356a base pytorch-onnx_1.3-
                                       069ea134-3346-5748-b513-
  py3.7-edt
  49120e15d288 basescikit-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 basepytorch_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-
                                              154010fa-5b3b-
py3.6
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
basetensorflow_2.1-py3.6
                                1eb25b84-d6ed-5dde-b6a5-
3fbdf1665666 base spark-mllib_3.2
20047f72-0a98-58c7-9ff5-a77b012eb8f5 base tensorflow_2.4-
py 3.8 \hbox{-horovod}\ 217 \hbox{c} 16 \hbox{f} 6 \hbox{-} 178 \hbox{f} - 56 \hbox{b} \hbox{f} - 824 \hbox{a} \hbox{-} \hbox{b} 19 \hbox{f} 20564 \hbox{c} 49\ \hbox{base}
runtime-22.1-py3.9-cuda
                                    26215f05-08c3-5a41-a1b0-
da66306ce658 base do_py3.8 295addb5-9ef9-547e-9bf4-
92ae3563e720 base
                          2aa0c932-798f-5ae9-abd6-
autoai-ts_3.8-py3.8
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-
                              49403dff-92e9-4c87-a3d7-
r_{3.6}
a42d0021c095 base xgboost_0.90-py3.6
4ff8d6c2-1343-4c18-85e1-689c965304d3 base pytorch-onnx_1.1-
                                    50f95b2a-bc16-43bb-
py3.6
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 displaymore use 'limit'parameter.
In [37]:
  import
  sklearn
  sklearn.___
  version_
Out[37]:
  '1.0.2'
In [38]:
 MODEL_NAME =
  'university'
 DEPLOYMENT_N
  AME = '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_spec
   s': [],'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': []}}
In [42]:
 model_id =
 client.repository.get_model_id(model_details)
 model\_id
 Out[42]:
 '85e3a479-6eb2-4104-9dd7-3c208aac046b'
In [43]:
 deployment_props = {
   client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
   client.deployments.ConfigurationMetaNames.ONLINE: {}
  }
 deployment =
    client.deployments.creat
    e(artifact_uid =
    model_id,
    meta_props = deployment_props
 )
Out [43]:
 ########################
```

Synchronous deployment creation for uid: '85e3a479-6eb2-4104-9dd7-3c208aac046b' started
######################################
initializing Note: online_url is deprecated and willbe removed in a future release. Use
serving_urls instead.ready
Successfully finished deployment creation, deployment_uid='b76c9fcd-b8a6-4fa6-bca7-ced79cce8301'
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>
        The modelhas predicted that you have chance
        <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">
```

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 preparingto get a betteridea.

```
<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"</pre>
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"</pre>
                name="tofel"min="50" max="120" required>
id="tofel"
                  </div>
                </div>
                <div class="row mb-3">
Rating:</label>
<label for="university_rating" class="col-lg-2 col-form-label">University
<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"</pre>
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">
                                               class="form-control"
                              type="number"
                 name="lor"step="0.01" min="1" max="5" required>
id="lor"
                   </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"</pre>
name="cgpa"step="0.01" min="5" max="10" required>
                  </div>
                </div>
                <fieldset class="row mb-3">
                   <legend class="col-form-label col-sm-2pt-0">Research:</legend>
                   <div class="col-sm-10">
```

```
<div class="form-check">
                        <input class="form-check-input" type="radio"</pre>
name="yes_no_radio"id="gridRadios1" value="1">
                        <label class="form-check-label"</pre>
                        for="yes_no_radio">Yes
                        </label>
                      </div>
                     <div class="form-check">
                        <input class="form-check-input" type="radio"</pre>
name="yes_no_radio"id="gridRadios2" value="0" checked>
                        <label class="form-check-label"</pre>
                        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...
                     </div>
                      <div class="spinner-grow text-primary m-1" role="status">
```

```
<span class="visually-hidden">Loading...</span>
</div>
</di>
```

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,</pre>
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 AdmitEligibility 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"</pre>
class="d-inline-blockalign-text-top">
         University AdmissionEligibility 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//f6V8Qbsw 3" crossorigin="anonymous"></script>
</body>
```

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>
         The modelhas predicted that you have no chance
         <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');
constloading = document.getElementById('spinner');
const disableButton = ()=>{
console.log('Submitting form...');
button.disabled = true;
button.className = "btnbtn-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);
}
```

REQUIRED IMAGES IN THE HTML CODES:









PYTHON CODE

App.py

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

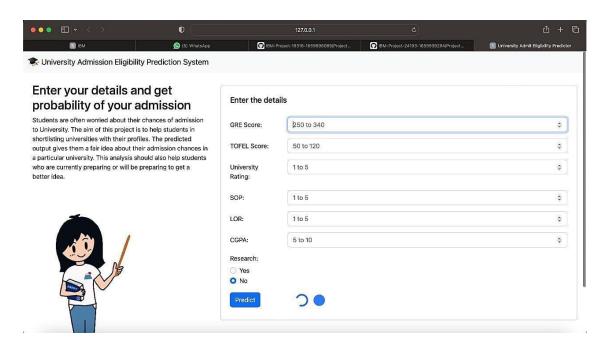
app = Flask(__name___)

@app.route("/", methods = ['POST', 'GET'])

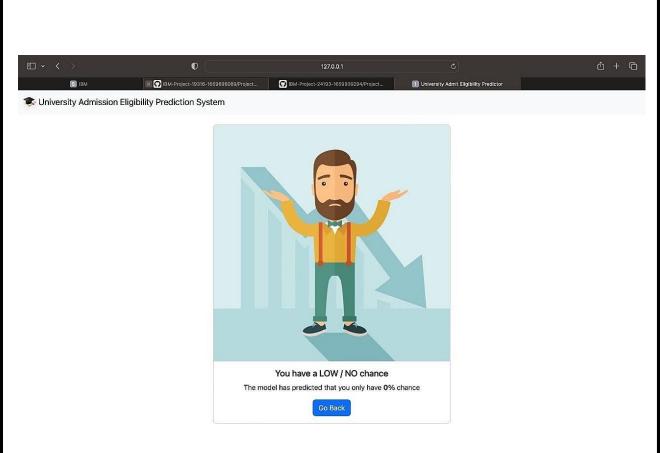
```
def index():
  if request.method == 'POST':
    arr = []
    for i in request.form:
      val = request.form[i]
      if val == ":
        return redirect(url_for("demo2"))
      arr.append(float(val))
    API_KEY = "Ct8qx91QEDmW9Xw2viaF7uGxI-jHrk0ogjKl17b_L4-K"
    token_response = requests.post('https://iam.cloud.ibm.com/identity/token', 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/b76c9fcd-b8a6-4fa6-bca7-
ced79cce8301/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:







12.2 GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-31934-1660206685

12.3	DEMO LINK:
	https://github.com/IBM-EPBL/IBM-Project-31934-
	1660206685/tree/main/Final%20Deliverables/video
	1000200003/tree/main/rmar/020Denverables/video