IBM – Nalaiaya Thiran

PROJECT DOCUMENTATION REPORT

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

Team ID: PNT2022TMID00994

Team members:

- D. Ganesh
- S. Akash
- E.A. Bharath
- S. Gurubaran

CONTENTS

INTRODUCTION

Project Overview

The primary concern on the minds of most passed out students is their university admissions. It is something which can greatly alter their future and hence they are often worried and anxious about it. Each and every university have their own admission process, and admit students based on the criteria set by them pertaining to the admission process. Unfortunately, most students are unaware of these criteria and this hinders their chances of getting admitted into those universities.

It'd be really helpful for the students if they can find out their chances of getting admitted into different universities since they can take decisions regarding admissions based on that. And the university admit eligibility predictor does just that. It predicts if the user has a chance of getting admitted into a certain university based on an 'eligibility score'. If the score is found to be above a threshold value, the user is informed that they are eligible.

The criteria considered by this specific model are: GRE score, TOFEL score, University ranking, SOP, LOR, CGPA, Research. An eligibility score is calculated based on these criteria which is used to decide if the user is eligible/ineligible. It has been ensured that the accuracy of model is as high as practically possible. Hence, this would be a handy tool for students aspiring to get admitted into prominent universities to find out their eligibility chances.

Purpose

The university eligibility predictor is a tool that is designed for students who are finished with their schooling/UG and aspire to get admitted into prominent universities. It provides them with their chances of admission in different universities based on certain criteria pertaining to their admission process. It could prove to be really helpful as it can provide some clarity to students on universities to which they can apply for admission.

The predictor would have served its purpose if it is found out that the eligibility chances predicted by it are close enough to actual results, since it'd mean that it is a trustable tool for students.

LITERATURE SURVEY

- **What is a university admit eligibility predictor?**

A university eligibility predictor is a tool that is designed for students who are finished with their schooling/UG and aspire to get admitted into prominent universities in India/abroad. It provides them with their chances of admission in different universities based on certain criteria pertaining to their admission process. It could prove to be really helpful as it can provide some clarity to students on universities to which they can apply for admission.

- **Problem statement**

How to design an eligibility predictor for students which provides them with their chances of getting admitted into different universities based on their scores and other important criteria.

USE CASE:

University education is becoming a critical pillar of social and economic life in the twenty-first century. It is crucial not only in the educational process but also in assuring two vital things: a great job and financial stability. On the other hand, might be extremely challenging because pupils are unaware of the admission standards. They waste lot of time and money.

Social Impact:

It will help students making decision for choosing a right college.

Business impact:

Getting potential students automatically business will increase

EXISTING SOLUTION I : CollegeAI

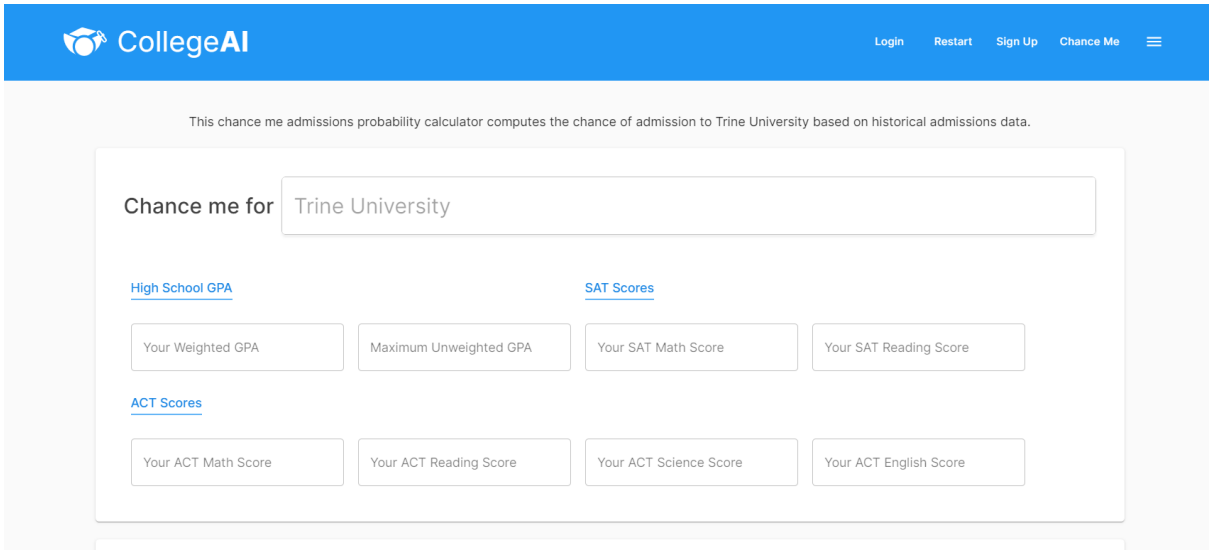
This chance me admissions probability calculator computes the chance of admission to Universities in US based on historical admissions data.

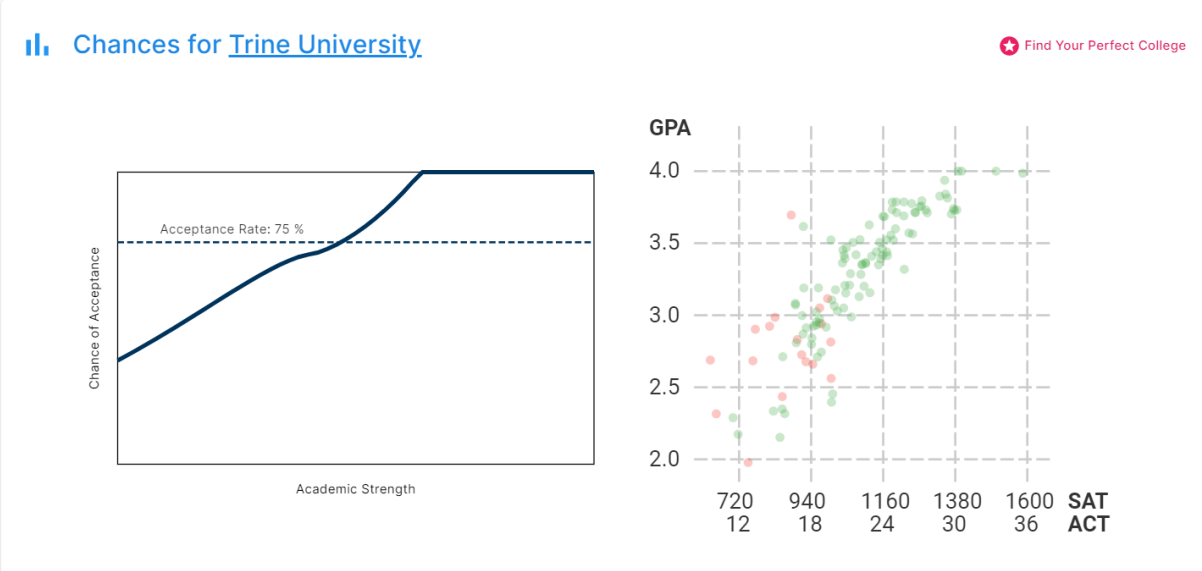
This predictor has been designed to predict chances of admissions for a student in 10 prominent universities in the United States (University of Arizona, Texas state University, Florida state university etc.)

It also has a 'Find your perfect college' feature which asks the user to fill out a small survey which questions them on their likes, dislikes, talents, skills and areas of interests and then suggest colleges which it finds suitable for the user based on the survey..,

Factors considered:

- High school GPA
- SAT score
- ACT score





Drawbacks:

- It covers only a limited number of universities
- Graphical representation would be a bit difficult to comprehend.
- It doesn't provide the user with their chances of getting a particular course at a specific university.
- Takes very few factors into account for judging eligibility.

Learnings:

The questionnaire employed to suggest colleges suitable to the student is a very interesting concept and might prove to be very beneficial to students. The UI is very smooth and easy on the eye from an user's POV.

EXISTING SOLUTION – II : CollegeVine

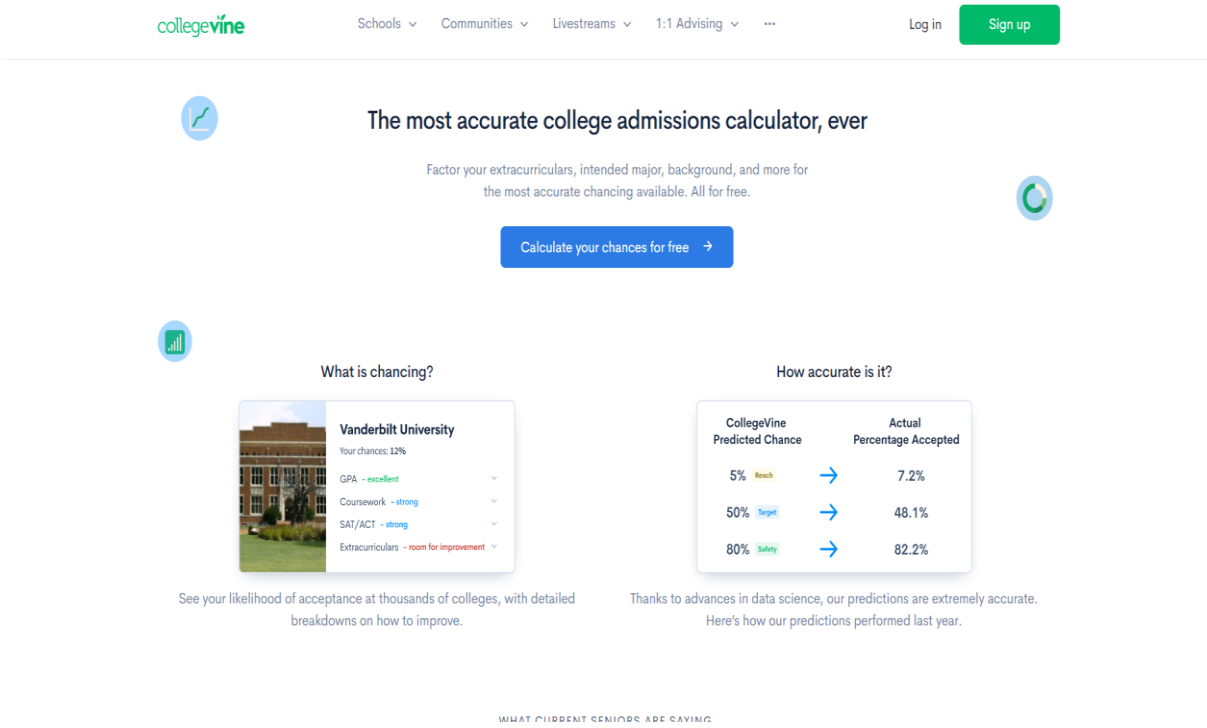
CollegeVine focuses on taking the guess work out of college admissions and clearing the misconceptions surrounding average acceptance rates and personal admission chances prevalent among students.

It also tries to take certain factors that might be outside of a student's control (diverse racial, economic and personal backgrounds) into account before deciding their chances.

It focuses on prominent universities based in the United States like Vanderbit university, Princeton university, Stanford university etc.

Factors considered

- Coursework
- GPA
- Test scores
- Extra Curriculars
- Demographics
- Intended major
- Class rank



Drawbacks:

- Not very user friendly in nature
- UI is not up to the mark and this might put the user through a lot of hassles
- Misuse of data submitted by the user is a possibility

Learnings:

The whole process behind calculating eligibility chances is explained in simple terms for the user's benefit which is a very important aspect from an user's POV.

This predictor takes a lot of factors into account including certain non-academic factors before judging the chances of a student which might be the reason behind their model's astounding accuracy rate.

EXISTING SOLUTION – III : Yocket

Yocket Admit Predictor gives the user the chances of making it into a particular program in a university they like. It is a very handy feature that's specifically designed for aspirants pursuing Masters in either US or Canada.

They have equipped it with Artificial Intelligence and Machine learning to give the user a close enough percentage based on the data information they have gathered along the way.

It is exclusively for students aspiring to get admitted into top universities outside of India.

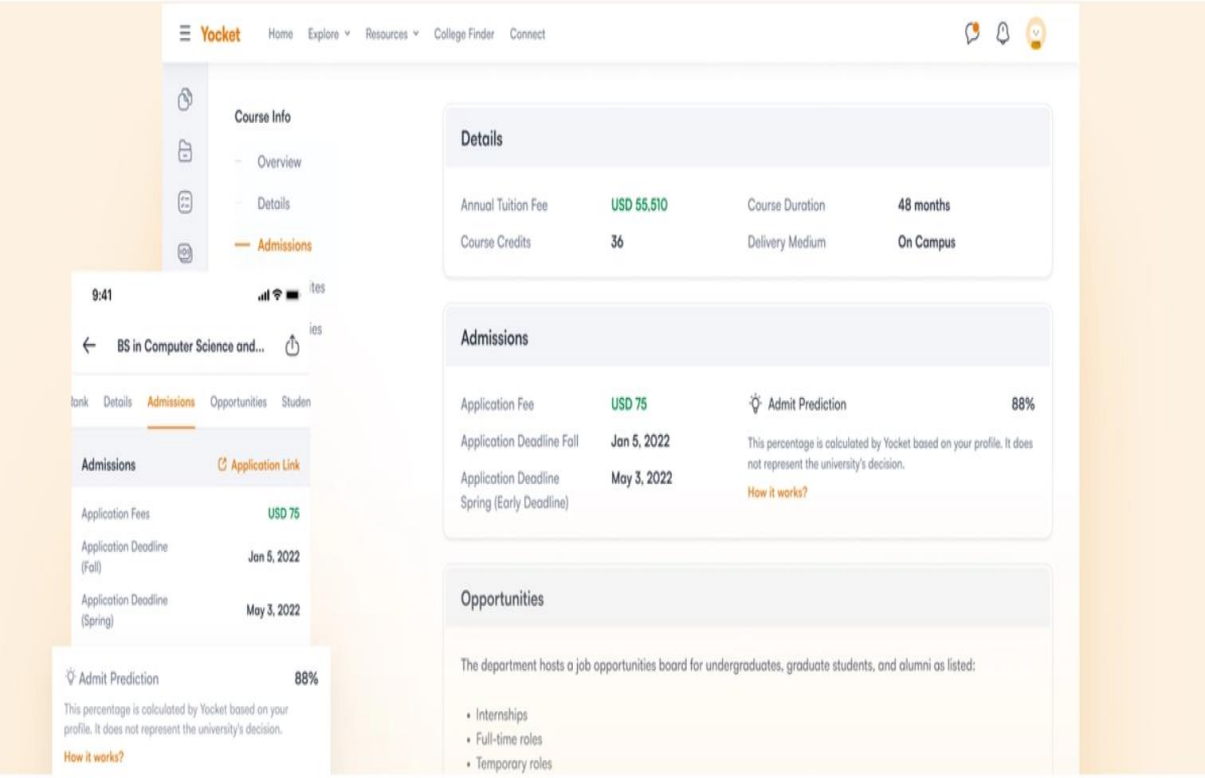
Admit Predictor works on the combination of historical Yocket user data, Artificial Intelligence based algorithms and Machine Learning.

Factors considered:

- GRE scores
- Aptitude scores
- UG scores
- English test scores
- Experience
- Technical papers

Drawbacks:

- Not free to use
- Restricted to Masters programs only
- Procedure is time consuming in nature and might test an user's patience



Learnings:

An algorithm has been developed by Yocket team which collects all the relevant data like admits, rejects, test scores, CGPA, course studied and more provided by previous Yocket users over the years. The algorithm has created a pattern of admits and rejects across courses and universities based on the collected information.

When the user adds information to their profile, it compares and calculates the admit chances based on the pattern it has deciphered.

The Machine Learning aspect of the feature takes in their information including admit and reject record to update its database and results in the future

EXISTING SOLUTION – IV : Careers360

It lets it users know their college admission chances based on their rank, home state, caste, etc

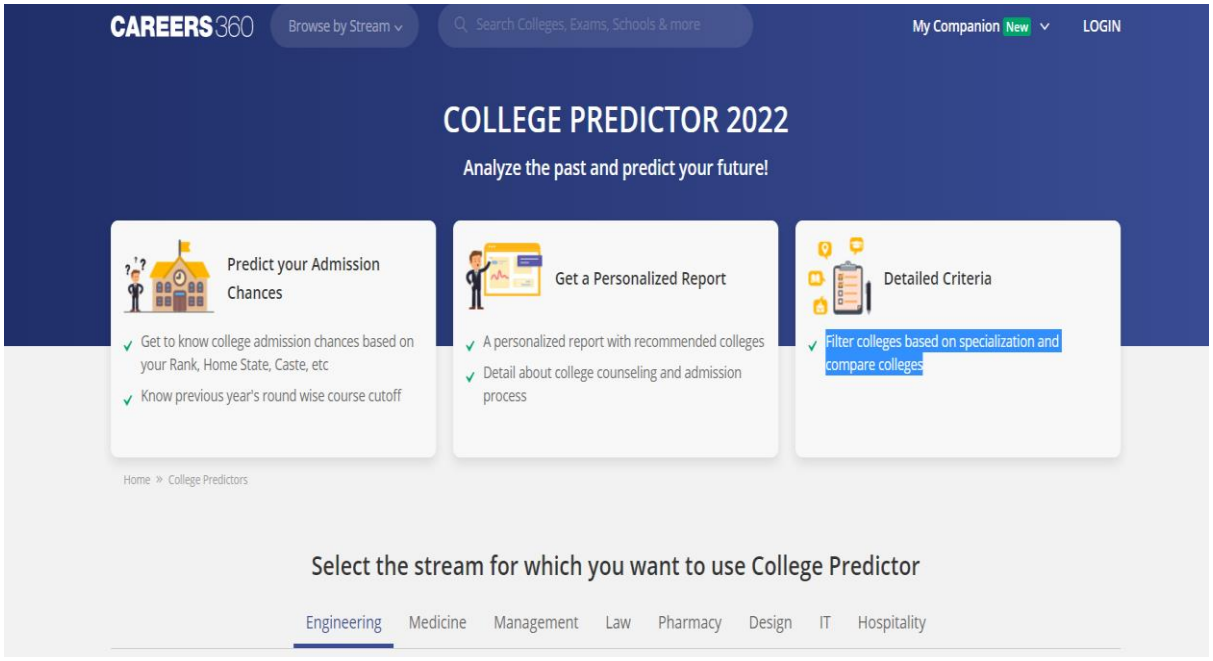
It additionally provides them with previous year's round wise course cut-off and details about the counselling and admission process

A personalized report with recommended colleges is designed exclusively for the users

Finally, it lets its users filter colleges based on specialization and compare colleges

Factors considered:

- Rank
- Home state
- 12^{th} marks
- Entrance exams scores
- Choice of course



Drawbacks:

- Not many factors are not taken into consideration
- This predictor is not reliable as other predictors found online
- There is no information about the working procedure of the algorithm used for this model.

Learnings:

This predictor covers a lot of colleges across varied fields (engineering, medicine, law, pharmacy etc..), since its scope is very large , it could prove to be the one-stop destination for all passed out students. Since it has to cover a lot of universities across diverse fields, it has to factor in different entrance exams specific to each of those fields into their algorithm.

INFERENCES

Upon carefully analysing these existing solutions and several other research papers related to our project, 'University admit eligibility predictor', we have made the following interferences which might aid us in our bid to design an eligibility predictor of our own:

I. We have to take into account as many factors as to increase the accuracy and reliability of the model.

II. The predictor should be user friendly in nature and the whole process should be hassle free.

III. The UI of the predictor should be smooth and decluttered.

IV. We need to try and increase the scope of our predictor as much as possible to increase the number of users accessing our product.

V. It should be ensured that the whole process doesn't turn out to be a the time-consuming affair for the user.

VI. We should try to convey the working procedure behind the model in simple terms to the user.

VII. Data collection is one of the most important steps in the process of designing the model and achieving a near perfect data set could go a long way in increasing the accuracy and reliability of our model.

VIII. Scores secured by the students in entrance exams should be taken into consideration as well since certain university base their admissions on those scores.

IX. Certain non -academic criteria could also be a deciding factor in the admission process of some universities, so we need to research them and factor it into our model if possible.

X. The privacy/security of the user should be of utmost importance and the data provided by them should be safe guarded.

REFERENCES

- CollegeAI predictor

  https://collegeai.com/chanceme

- CollegeVine predictor

  https://www.collegevine.com/admissions-calculator

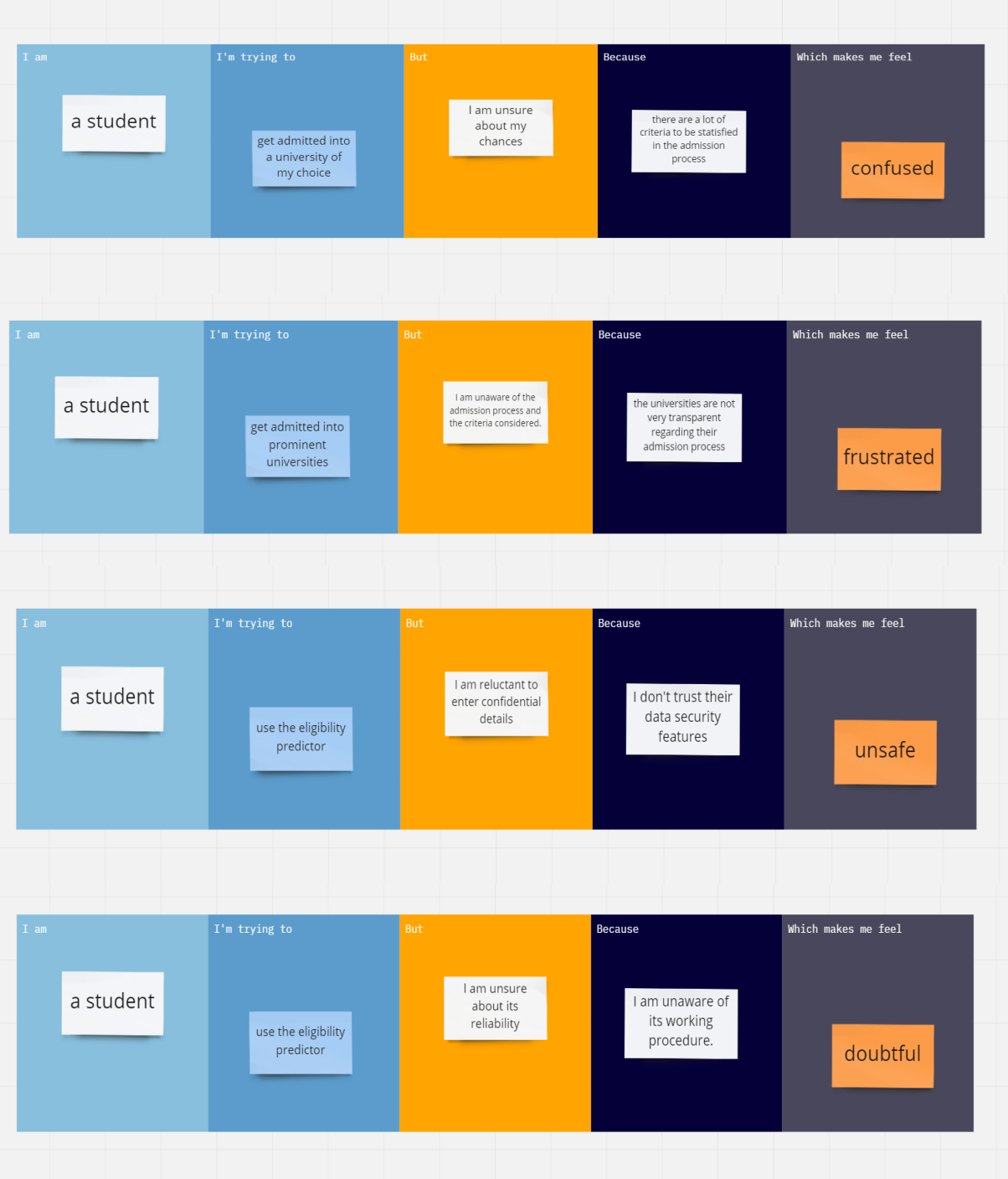
- Yocket's admit predictor

  https://yocket.com/blog/yocket-admit-predictor

- Careers360 predictor

  https://www.careers360.com/college-predictor

PROBLEM STATEMENT DEFINITION

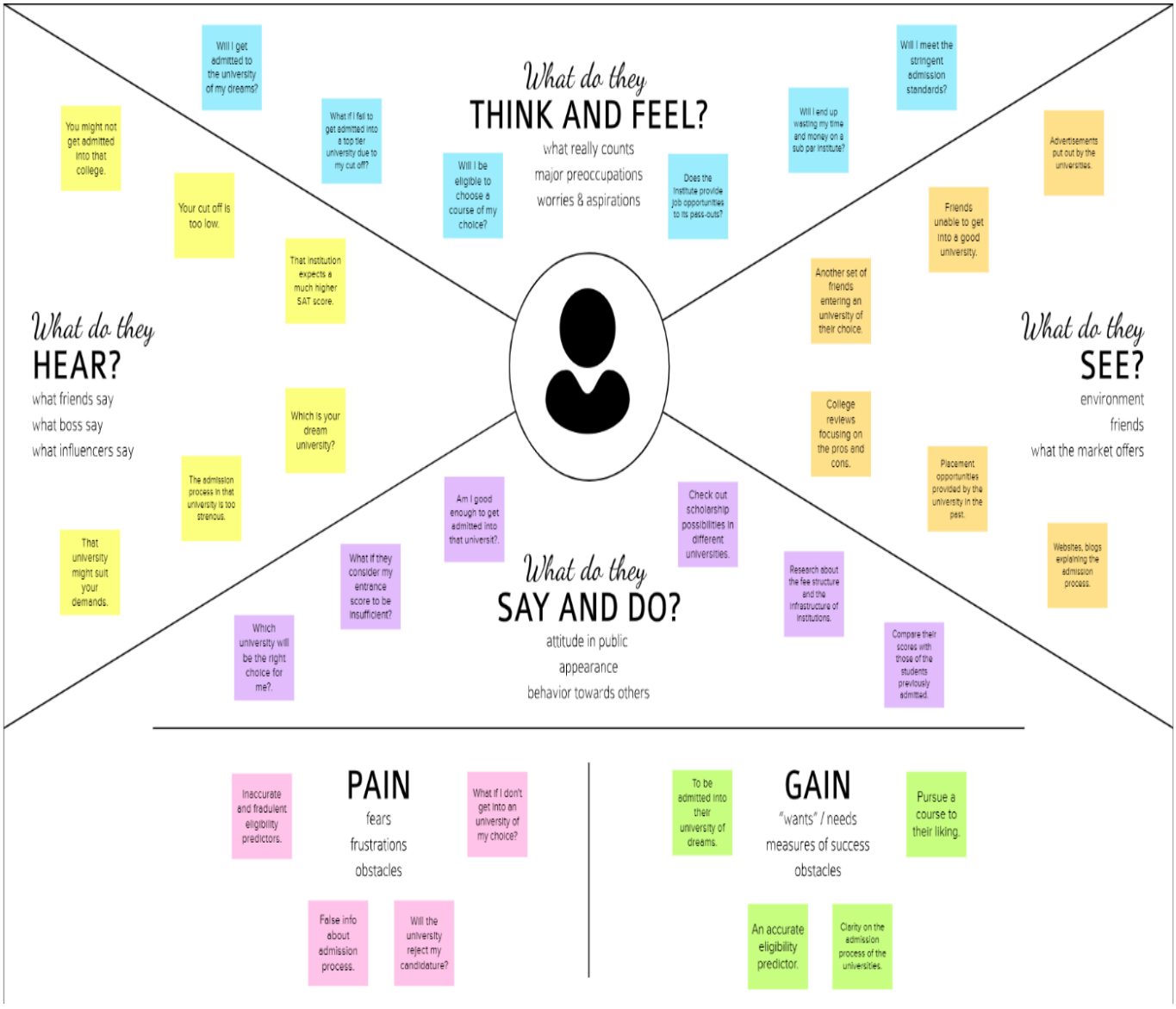


Customer Problem Statement

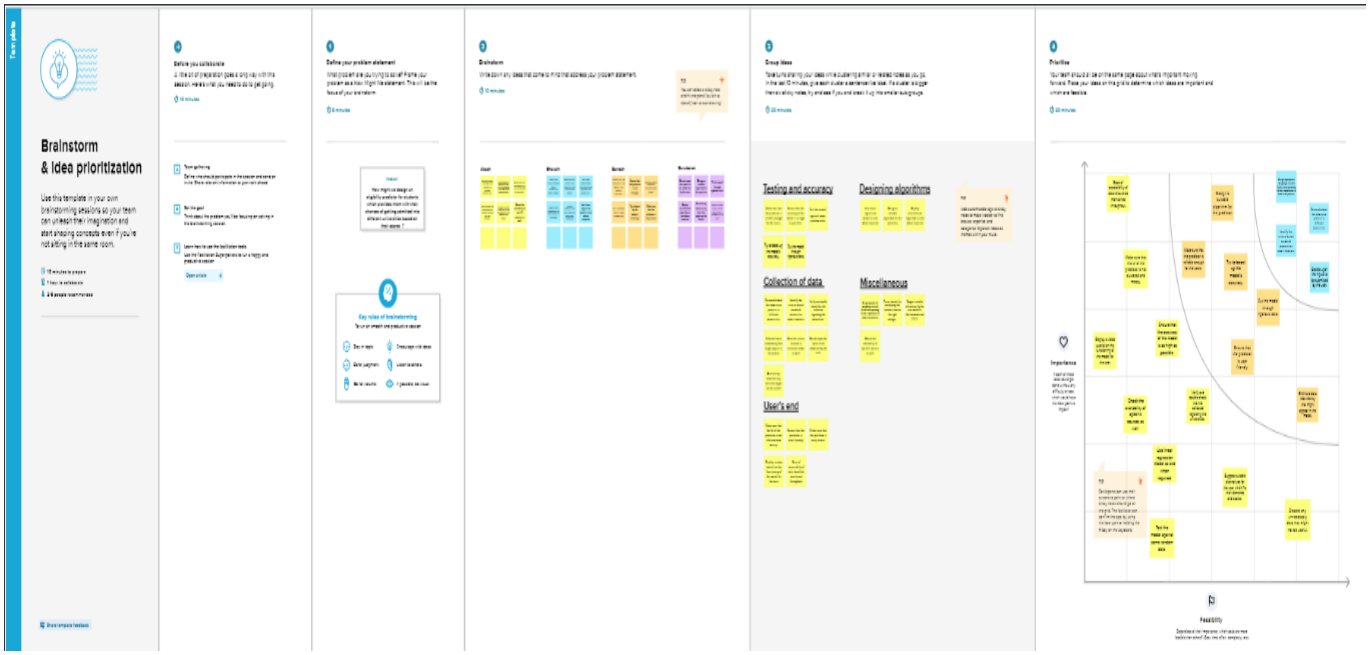
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A student	To get admitted into a university of my choice	I am unsure about my chances.	There are a lot of criteria to be satisfied in the admission process.	confused
PS-2	A student	To get admitted into a prominent university	I am unaware of the admission process and criteria considered.	Universities are not very transparent regarding the admission process	frustrated
PS-3	A student	Use the eligibility predictor	I am reluctant to enter confidential details.	I don't trust its data security features	unsafe
PS-4	A student	Use the eligibility predictor	I am unsure about its reliability.	I am unaware of its working procedure	doubtful

IDEATION & PROPOSED SOLUTION

EMPATHY MAP CANVAS



IDEATION AND BRAINSTORMING

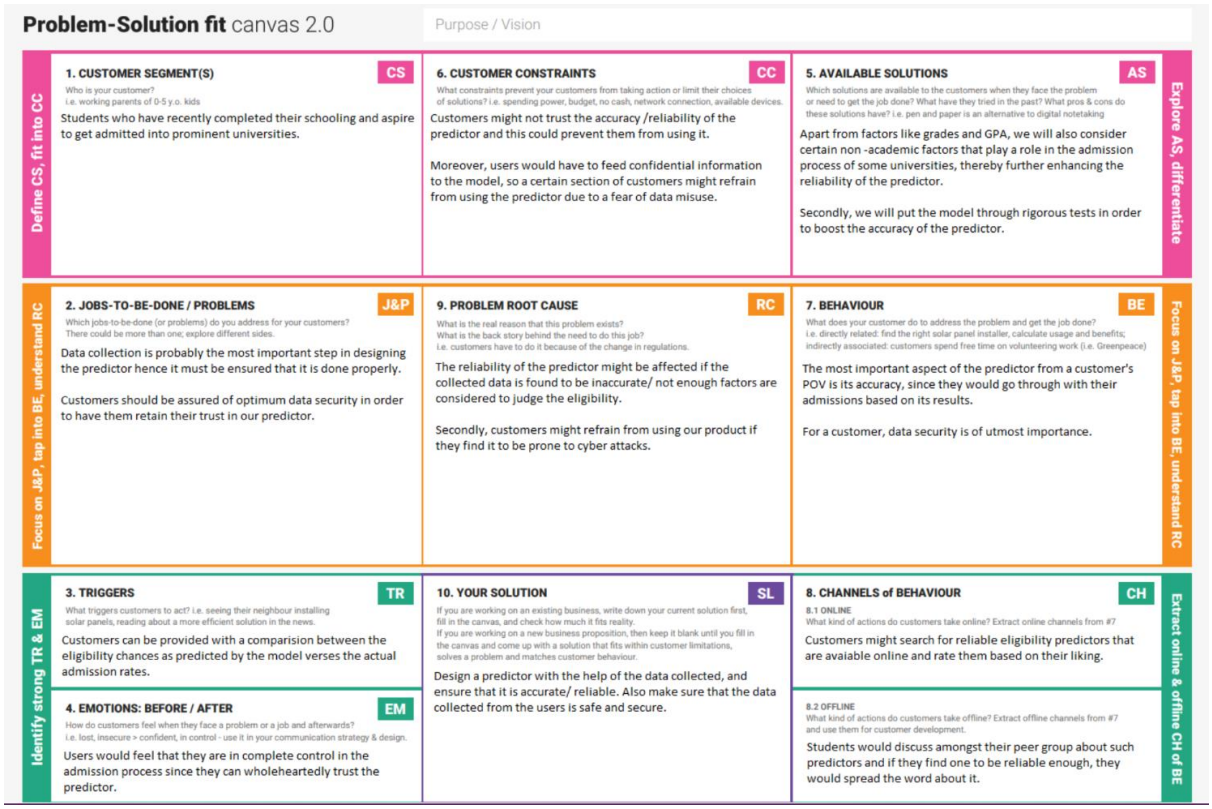


PROPOSED SOLUTION

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	How might we design an eligibility predictor for students which provides them with their chances of getting admitted into different universities based on their scores and other important criteria.
2.	Idea / Solution description	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 regarding the admission process. A model is developed which analyses the data provided by the user and evaluates it in accordance with the algorithm developed to predict the eligibility of the user for the specified university.

3.	Novelty / Uniqueness	We aim to design the model in such a way that it takes certain non-academic factors which influence the admission process into consideration as well. This further enhances the accuracy of the predictor. This attribute is not considered in most predictors available in the market.
4.	Social Impact / Customer Satisfaction	This predictor would provide a clarity to passed out students who might be confused regarding their future with respect to university admissions. The students can apply to universities based on their eligibility chances.
5.	Business Model (Revenue Model)	Such predictors have a huge demand in the market since students who complete their schooling are always in need of tools like this to plan out their university admissions.
6.	Scalability of the Solution	The scope of this predictor is very wide as a large number of universities could be brought within the range of this predictor depending on the requirements of the user. Hence, this solution is largely scalable in nature.

PROBLEM SOLUTION FIT



REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Facebook
FR-2	Authentication of user	An OTP is sent to the registered phone number and email to authenticate the user.
FR-3	User Data (input) User confirmation	A confirmation mail/SMS is sent to the user after the successful registration
FR-4	User Data (input)	Details like CGPA, IELTS/TOEFL score, projects done, GRE score are collected from the user
FR-5	Log in/Log out	Users can login using their mail id and password. They can logout as and when required.
FR-6	Editing user profile	The users must have an option to edit their profile even after the initial registration is over.
FR-7	Chat box facility	A chat box to provide the answers to FAQs and resolve any issues in the functioning.
FR-8	Video tutorial	A video tutorial explaining the working of the predictor should be made available for the convenience of the user.
FR-9	Previous admission records	Admission records of the universities in the years before the current academic year, should be made available to the user.

Non-functional Requirements:

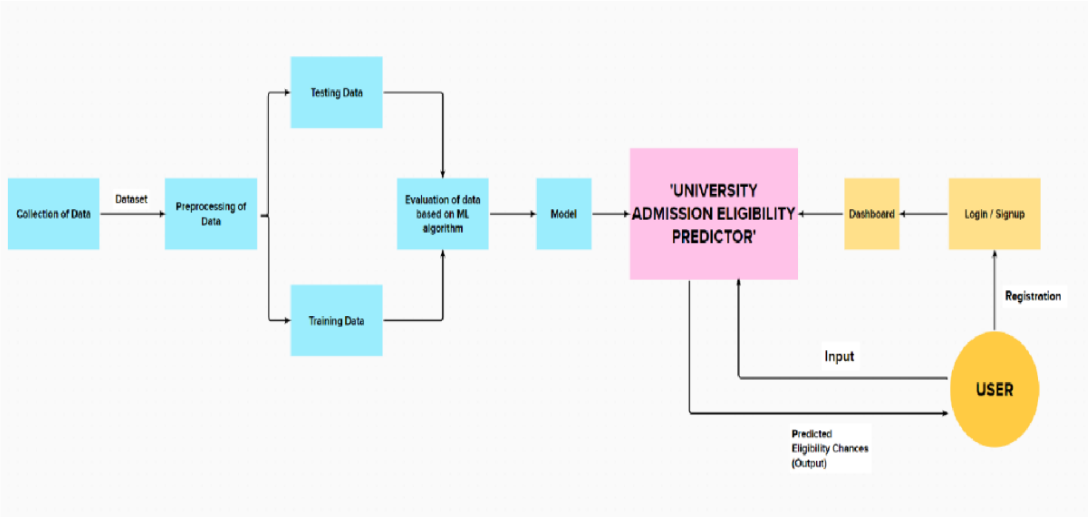
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The predictor must be easy to use and the UI should be smooth and decluttered.
NFR-2	**Security**	It should be ensured that necessary security features are in place to safe guard users' data from activities like data theft
NFR-3	**Reliability**	The reliability of the predictor must be maintained by providing the customer close-to-accurate results every single time.

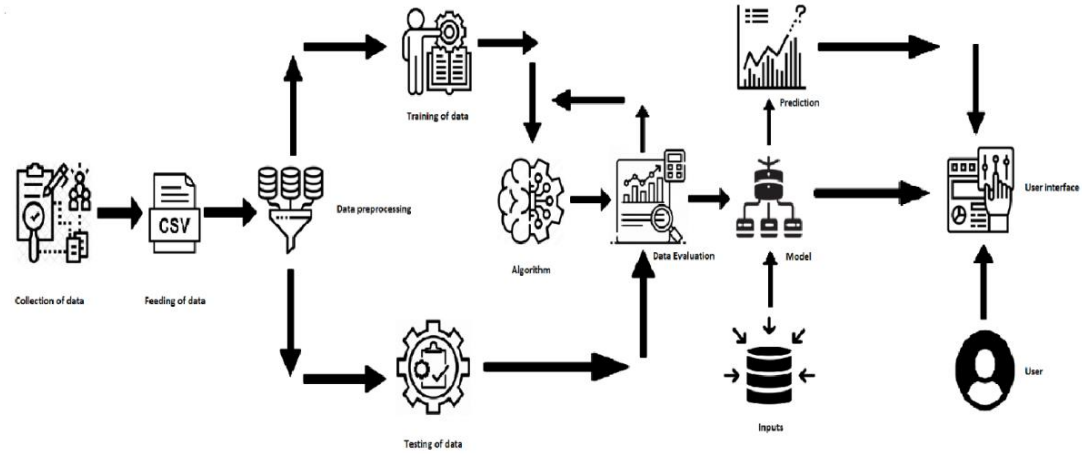
NFR-4	**Performance**	The performance of the predictor is entirely dependent on its accuracy and the time taken by it to come up with the results.
NFR-5	**Availability**	It must be made accessible through any browsers to ensure that it is available to a wide spectrum of users.
NFR-6	**Scalability**	The predictor must be designed in such a way that its range/scope can easily be increased without any massive changes
NFR-7	**Serviceability**	Customer service must be provided through chat box/chat bots to resolve any issues that they might face and to resolve their queries.
NFR-8	**Manoeuvrability**	The platform must be easily manoeuvrable.

PROJECT DESIGN

DATA FLOW DIAGRAM



SOLUTION ARCHITECTURE



USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user/Web user)	Registration (Sign Up)	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail or Facebook	I can register & access the dashboard with Facebook or Gmail	Medium	Sprint-2
		USN-4	A disclaimer explaining our policy against unfair promotion of any university is displayed before registration.	I read the disclaimer and then proceed with the registration process.	Low	Sprint-1
	Login/Signup	USN-5	As a user, I can log into the application by entering email & password	I am able to log in to the application using my email and password.	High	Sprint-1
	Dashboard	USN-6	As a user, I can view my personal information on the dashboard	I am able to see my personal info on the dashboard	Low	Sprint-1
		USN-7	I am able to access admission records of different universities previously.	I can access the files containing the records but with read only permission.	Medium	Sprint-2
Customer Care Executive	Chatbot/Chat box	USN-8	As the customer care executive, I am responsible for responding to the	I can access the chatbot/chatbot	Low	Sprint-3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			user's queries.			
Administrator	Data collection and management.	USN-9	As the admin, I am responsible for updating the predictor regularly based on any changes in the universities' admission process	I am able to update the changes successfully	High	Sprint-2
		USN-10	I have complete access and control over the resources of the model.	I have complete control over the predictor.	Medium	Sprint-1
		USN-11	I have to resolve any issues that might arise while using the model.	I can resolve the issues that arise while using the predictor.	Low	Sprint-3
		USN-12	I must make necessary changes to the model as and when required based on customer feedback	I am able to makes necessary changes to the model without altering its functionality	High	Sprint-3

TECHNICAL ARCHITECTURE

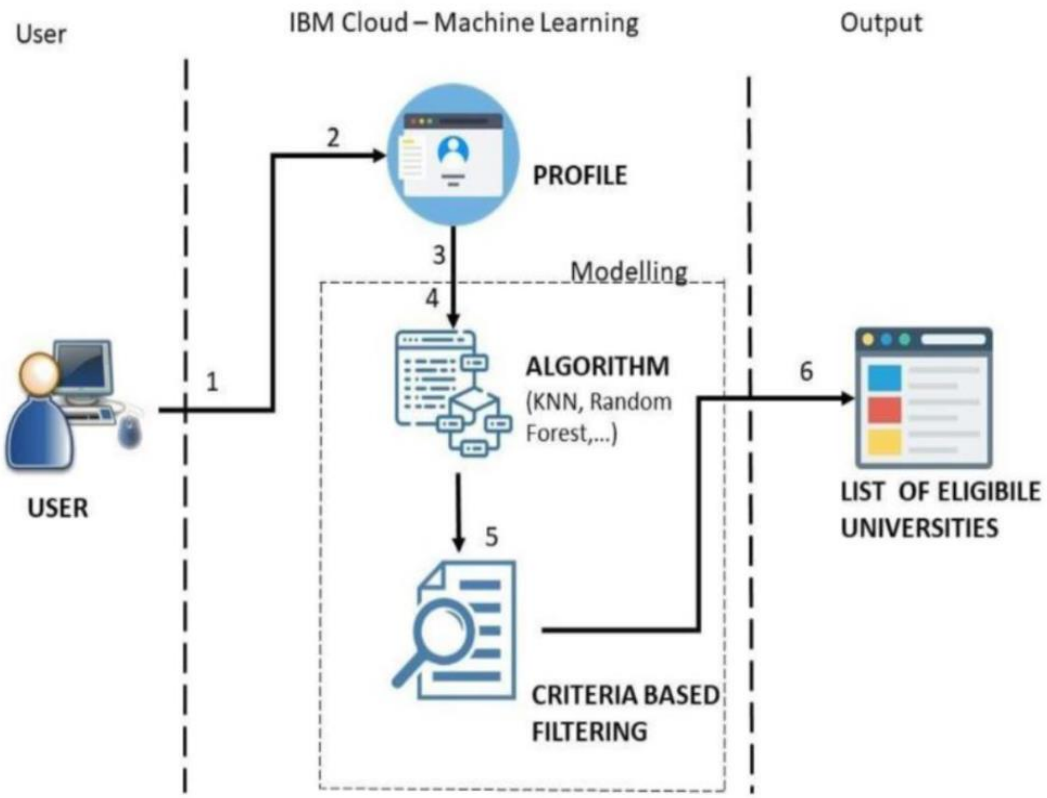


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	How user interacts with application and its features.	HTML, CSS, JavaScript etc.
2	Application Logic-1	The user fills the data into his profile which is then fed into the model to calculate the chances	Python [Jupyter]
3	Application Logic-2	The model predicts the eligibility chances of the user for different universities based on the input data	IBM Watson STT, Python

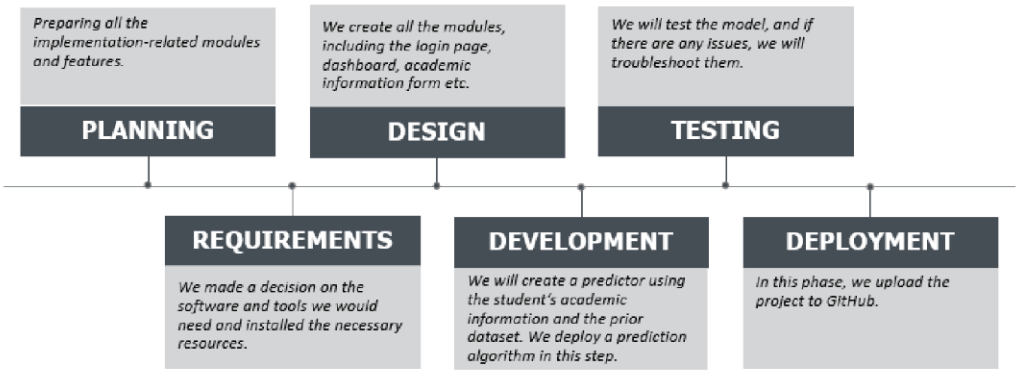
4	Database	Data of the names of the universities and their corresponding cut-offs and exam scores for admission	Imported through pandas in a csv format
5	Machine Learning Model	Predicts the output using the ML algorithm	KNN, Decision tree, Random Forest, etc.
6	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	IBM cloud, local cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python for backend and Flask for front end	Python, Flask
2.	Security Implementations	To ensure the security of the data provided by the user	Encryption, OWASP
3.	Scalable Architecture	The model is scalable in nature because its scope can be increased easily.	Random forest ML algorithm, Logistic regression
4.	Availability	The model is available to anyone, anywhere, anytime	IBM load balancer
5.	Performance	The chances are predicted with a greater accuracy	Random forest ML algorithm

PROJECT PLANNING & SCHEDULING

MILESTONE AND ACTIVITY LIST



Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Ganesh, Bharath, Akash, Gurubaran
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application.	1	High	Ganesh, Bharath, Akash, Gurubaran
Sprint-2		USN-3	As a user, I can register for the application through Facebook and Gmail.	2	Low	Ganesh, Bharath, Akash, Gurubaran
Sprint-2	Authentication	USN-4	An OTP is sent to a registered phone number and email id to authenticate the	2	Medium	Ganesh, Bharath, Akash, Gurubaran

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			user.			
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Ganesh, Bharath, Akash, Gurubaran
Sprint-3	Dashboard	USN-6	As a user, I can view my personal information in the dashboard	1	Medium	Ganesh, Bharath, Akash, Gurubaran
Sprint-3	Data collection and Management	USN-7	As the Admin, I am responsible for updating the predictor regularly based on any changes in the universities' admission process	3	High	Ganesh, Bharath, Akash, Gurubaran
Sprint-4	Customer services	USN-8	As the customer service executive, I'm responsible for responding to the user's queries.	2	Low	Ganesh, Bharath, Akash, Gurubaran
Sprint-4	Feedback	USN-9	As a user, I can provide my feedback for the improvement of the application	2	Low	Ganesh, Bharath, Akash, Gurubaran

Project Tracker, Velocity & Burndown Chart

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

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

AV = Sprint duration / Velocity

= 6 / 4

= 1.5

CODING & SOLUTIONING

Demo:

```
<!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>

   {% 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>
{% extends 'index.html' %} {% block body %}
<div class="main">
   <nav class="navbar navbar-expand-lg bg-light">
      <div class="container-fluid">
         <a class="navbar-brand text-responsive-h" href="/">
```

```
            <img src="/static/img/logo.jpg" alt="Logo" width="30" height="50" class="d-inline-
block align-text-top "> UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
          </a>
        </div>
    </nav>
    <div class="MAIN">
        <p class="text-responsive,">
          This university eligibility predictor is a tool that is designed for students who are
finished with their schooling/UG and aspire to get admitted into prominent universities . It
provides them with their chances of admission in different universities
          based on certain criteria pertaining to their admission process. It could prove to be
really helpful as it can provide some clarity to students on universities to which they can
apply for admission.</p>
        <p>The aim of this project is to help students in shortlisting universities with their
profiles.
        </p>
    </div>
</div>
<!-- </div> -->
<div class="p-4">
    <div class="row mb-2">
        <div class="col-4">


        </div>


        <div class="row-6">
            <div class="card p-2 ms-2 my-2">
                <div class="card-body">
                    <h5 class="card-title pb-4">
                        Enter the details
                    </h5>
                    <div class="form">
                        <form action="/" method="post" class="form-horizontal" 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>
  </div>

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

‘Chance’ page:

```
{% 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/sucess.gif" height="30px" width="30px" class="card-img-top"
alt="...">
      <div class="card-body">
        <h5 class="card-title">Congratulations!</h5>
        <p class="card-text">Yay!The model has predicted that you have a chance</p>
        <p> Eligibility Score:<strong>{{content[0]}}%</strong> </p>
        <a href="/home" class="btn btn-primary">Go Back</a>
```

```
        </div>
      </div>
   </div>
</div>

{% endblock %}
```

‘No Chance’ page:

```
{% 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/faliure.gif" width="120px" height="auto" class="card-img-top"
alt="...">
         <div class="card-body">
            <h5 class="card-title">Sorry</h5>
            <p class="card-text">Oops!The model has predicted that you don't have a
chance</p>
            <p>Eligibility Score:<strong>{{content[0]}}%</strong> </p>
            <a href="/home" class="btn btn-primary">Go Back</a>
         </div>
      </div>
   </div>
</div>

{% endblock %}
```

CSS:

```
* {
   margin: 0;
   padding: 0;
   border: 0;
}

.heading {
   text-align: center;
   color: rgb(65, 199, 220);
}

.MAIN {
   padding-top: 20px;
   padding-left: 20px;
   padding-right: 20px;
```

```
    font-family: 'Times New Roman', Times, serif;
    font-size: 25px;
}

.navbar {
    text-align: center;
}

body {
    font: 62.5%/1.5 "Lucida Grande", "Lucida Sans", Tahoma, Verdana, sans-serif;
    background: #a3a9bde9;
    background: -webkit-linear-gradient(to right, #bccce9, #b6cef0);
    background: linear-gradient(to right, #bacbed, #bcd0ed);
    color: #000000;
    text-align: center;
}

h1 {
    font-size: 2.2em;
}

h2 {
    font-size: 2.0em;
}

h4 {
    font-size: 1.6em;
}

input.text {
    padding: 3px;
    border: 1px solid #999999;
}

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

.text-responsive-h {
    font-size: calc(80% + 0.6vw + 0.6vh);
}
```

Java Script file:

```
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;
```

Flask Interfacing code:

```
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 = "M9HzyQsT95Fg-UVXiBtaNwNEl_8vdomPyIOtKwTlPohb"
      token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
```

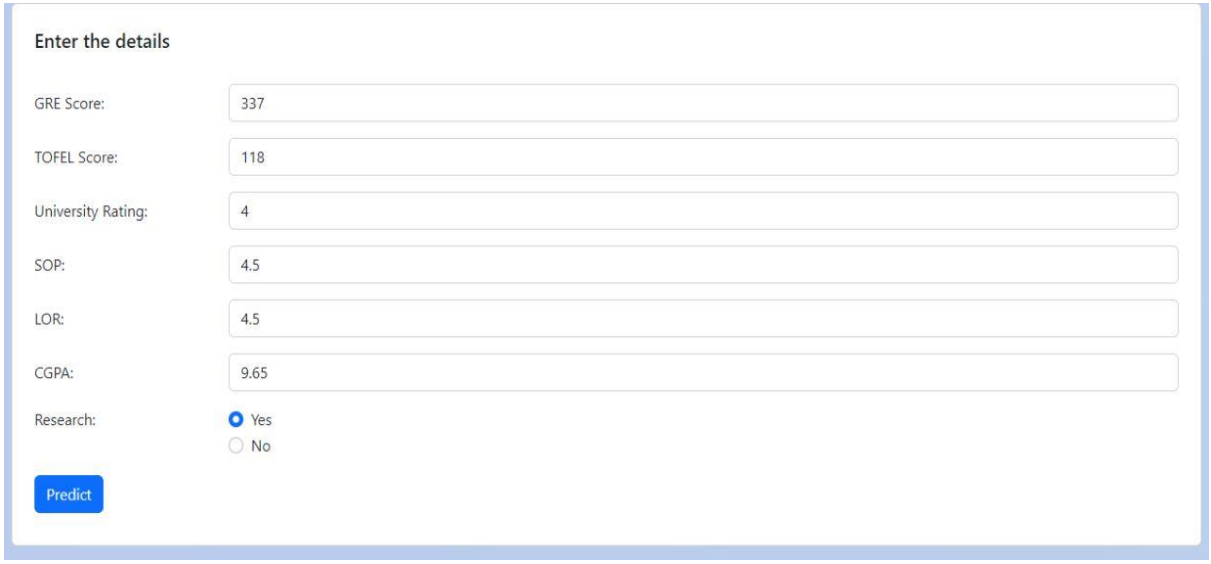
```
        "apikey": API_KEY,
        "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
        })
    mltoken = token_response.json()["access_token"]
    header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
    payload_scoring = {
      "input_data": [{"fields":[  '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/fcb08da1-ca92-4142-aab9-
6af6797d177c/predictions?version=2022-11-13',
      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():
```

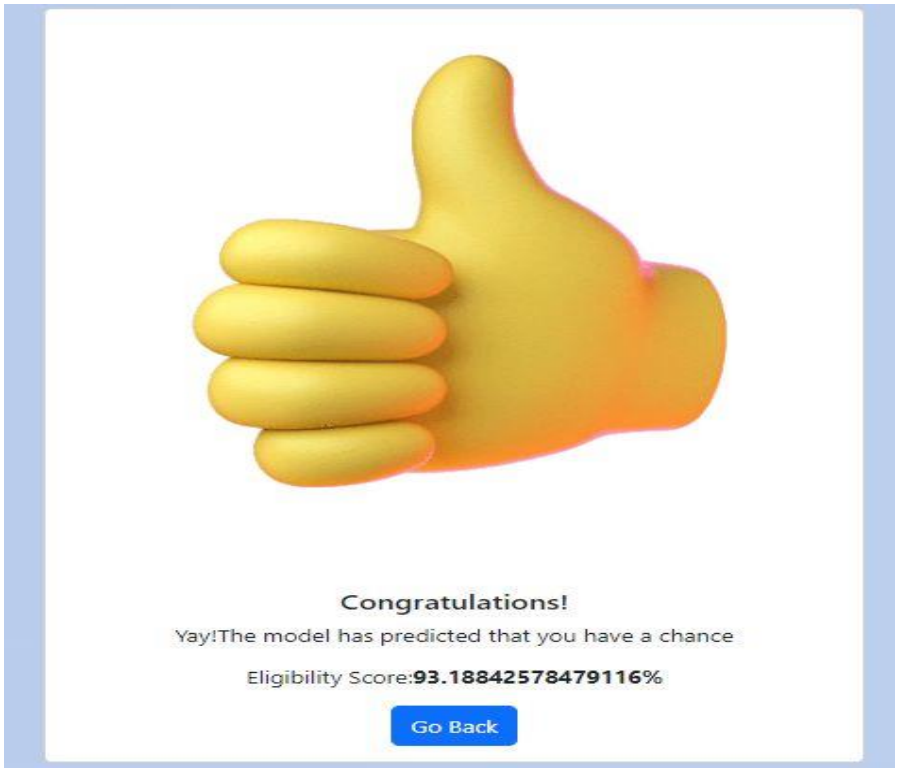
```
    return redirect(url_for("demo2"))

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

TESTING

Test cases (inputs from the dataset)

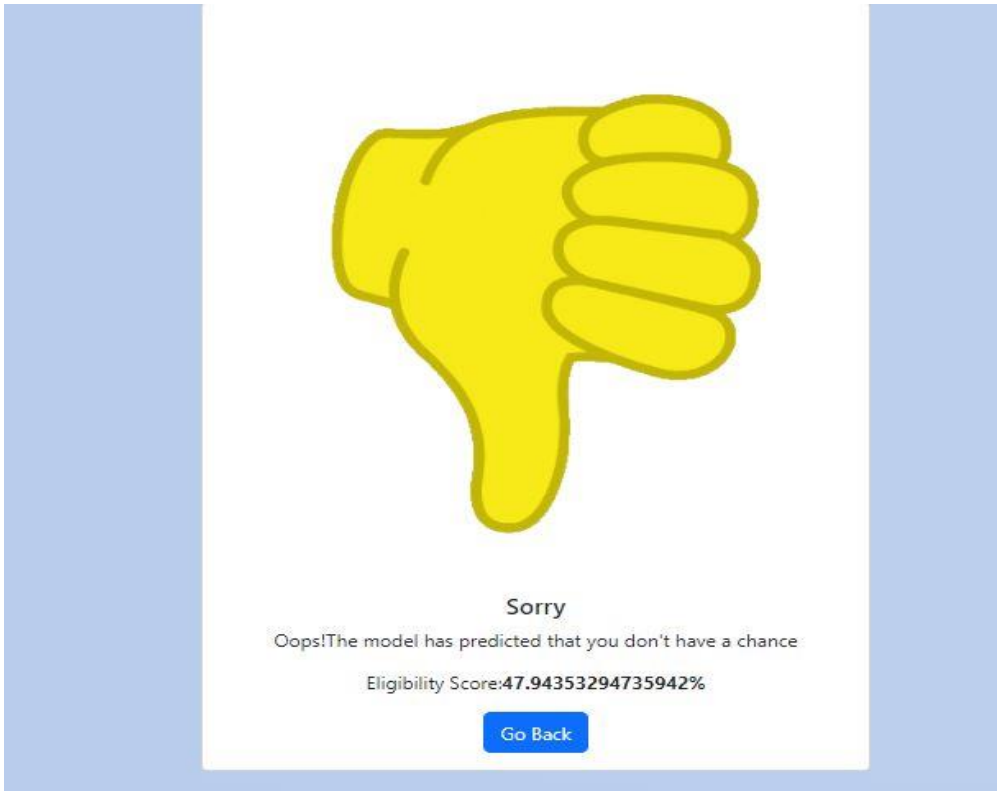
Case (i)





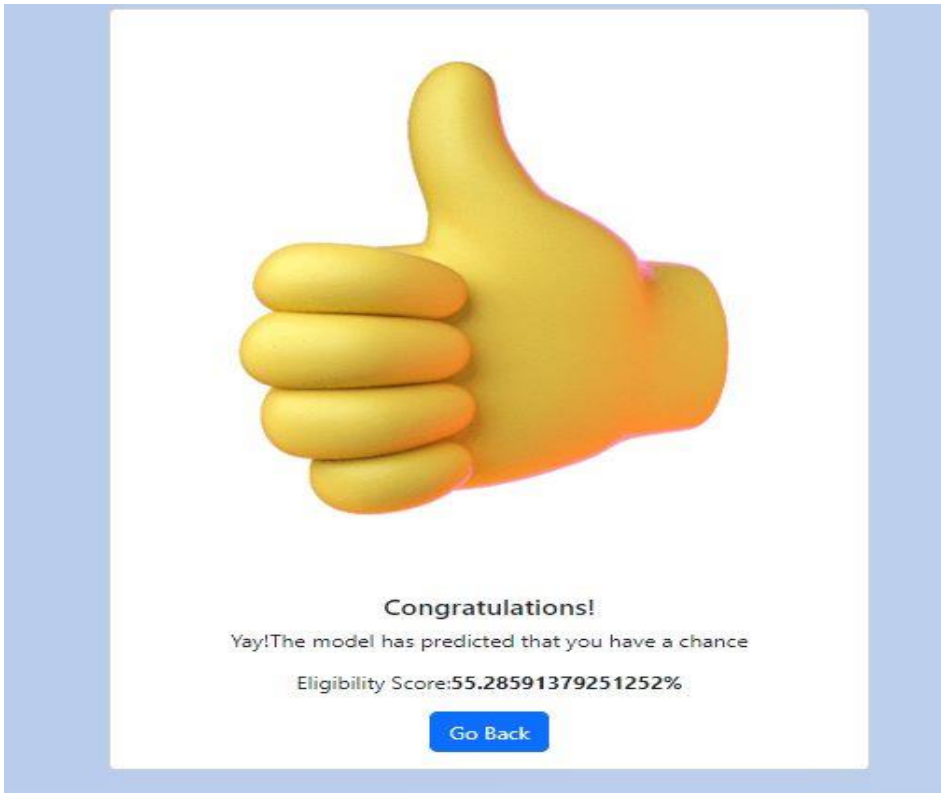
Case (ii)


Enter the details
GRE Score:
299
TOFEL Score:
80
University Rating:
1
SOP:
3
LOR:
2.7
CGPA:
7.88
Research:
Yes
No
Predict

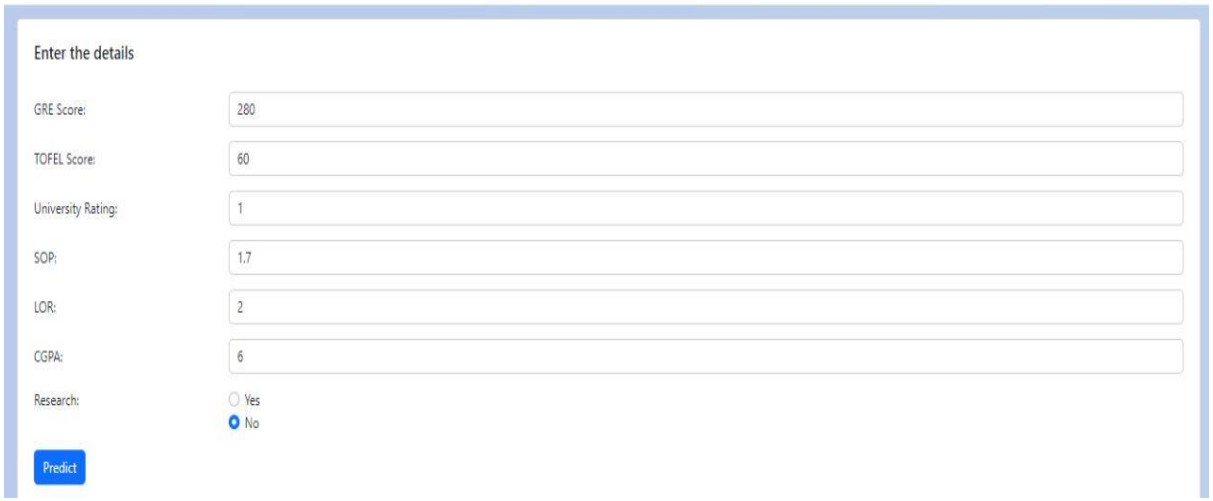

Sorry
Oops!The model has predicted that you don't have a chance
Eligibility Score:47.94353294735942%
Go Back

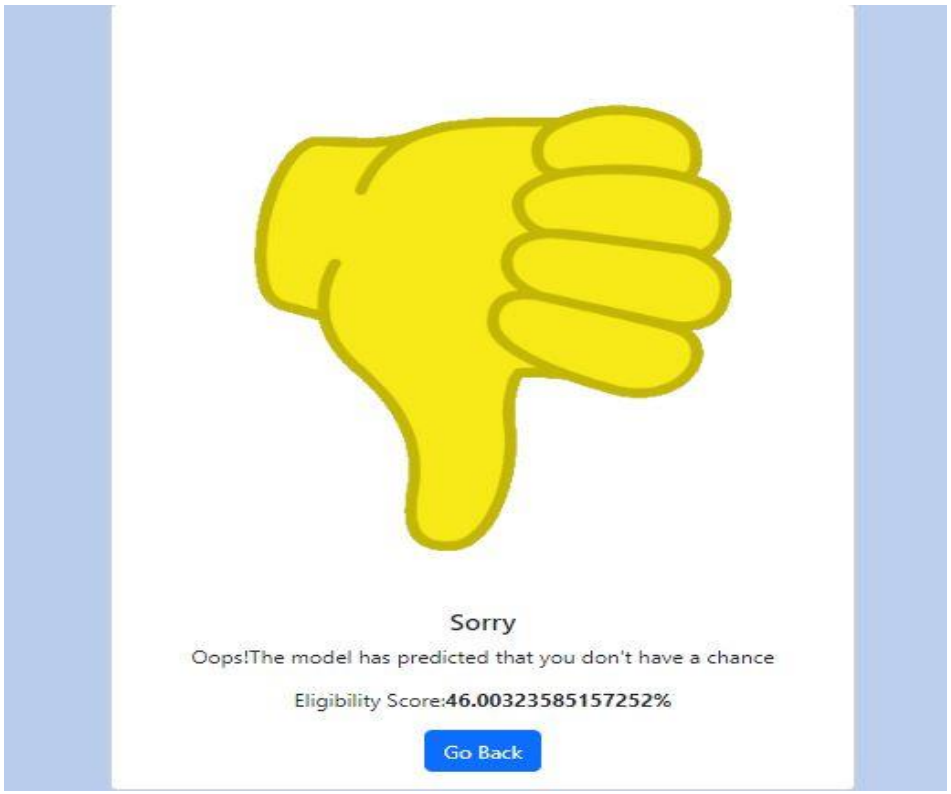
User acceptance testing

Case (i)



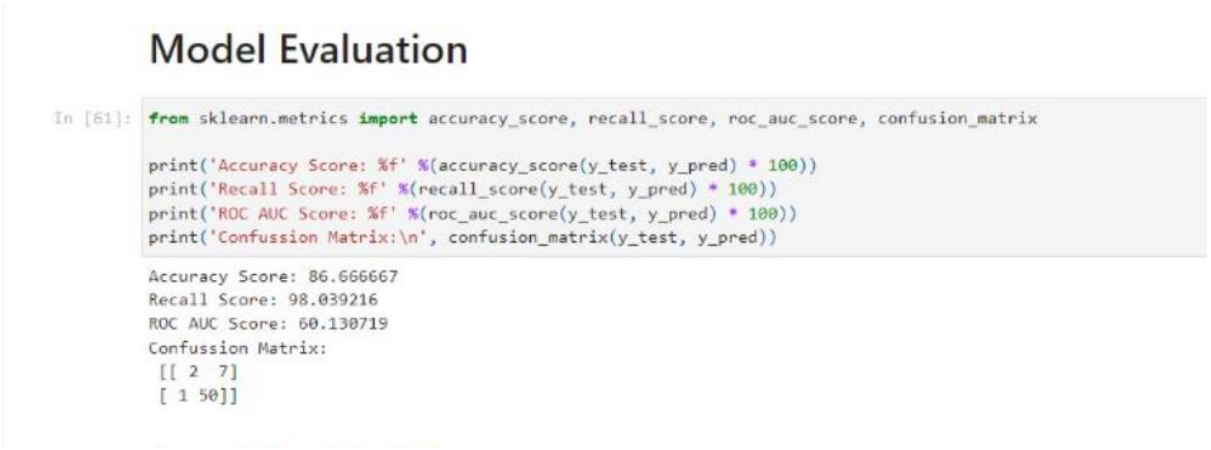
Case (ii)


Enter the details
GRE Score:
280
TOFEL Score:
60
University Rating:
1
SOP:
1.7
LOR:
2
CGPA:
6
Research:
Yes
No
Predict


Sorry
Oops!The model has predicted that you don't have a chance
Eligibility Score:46.00323585157252%
Go Back

RESULTS

Performance metrics



Accuracy score = 86.66

- ✓ It is the ratio of true positives and true negatives to all positive and negative observations.

Recall Score = 98.03

- ✓ It is used to measure the model performance in terms of measuring the count of true positives in a correct manner out of all the actual positive values.

ROC AUC score = 60.13

- ✓ It is a performance measurement for classification of problems at different threshold settings

Confusion matrix
[[2 7]
[1 50]]

- ✓ It is a matrix used to determine the performance of the classification model for a given set of test data

ADVANTAGES & DISADVANTAGES

Advantages:

- It helps student for making decision for choosing a right college.
- It is quick, accurate, efficient and reliable.
- It is very user friendly.
- This model avoids data redundancy and inconsistency.
- It provides the user with an eligibility score.

Disadvantages:

- Since the model is built on a limited dataset, the accuracy of the predictor can be affected as a whole.
- There are a number of non-academic factors that can determine whether the students get admitted into the university but the model doesn't take those into account.
- An active internet connection is required to access this predictor.

CONCLUSION

To conclude, we'd have to admit that this project has been more taxing than we had imagined before we began working on it. But it has been a huge and valuable learning curve for all of us as we were able to put all our theoretical knowledge to use in a practical scenario for the first time. We had to learn and unlearn a lot of things during the process of developing the predictor and we thoroughly enjoyed that aspect of it. We have put in a lot of effort into increasing the accuracy of the model. Our primary focus beyond boosting the accuracy was on developing a user-friendly model which has a clean and decluttered UI. We believe that we have achieved that. This project has a lot of scope and could be scaled if the demand is high in the future.

FUTURE SCOPE

Scalability of the model is one of the most important non-functional requirements which should be in focus while developing any model as it ensures that its scope is wide. Hence, we ensured that the predictor we developed is scalable in nature in order to meet any future demands. The user-friendly nature of the predictor will make sure that it is popular among students over an extended period of time. The predictor can be accessed anywhere anytime as long as the user has a stable internet connection. Advanced security features can be implemented in the model in the future to overcome data theft. There is always a possibility of increasing the accuracy score of the model by fine tuning the code. Hence, the predictor has a wide scope and can be developed in the future as per the customer's demands.

APPENDIX

Source code:

```
import pandas as pd
import numpy  as np
import matplotlib.pyplot as plt
import seaborn as sns
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your
credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
   ibm_api_key_id='MwLMvoaVaULL9-J0Yma10fFGdKwGGy0psJIkbuyuIJJP',
   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 = 'universityadmiteligibilitypredict-donotdelete-pr-fm8fmnu4arxxzo'
object_key = 'Admission_Predict.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head()
df.drop(['Serial No.'],axis=1,inplace=True)
df.describe()
df.info()
df.isnull().any()
df.isnull().sum()
sns.displot(x=df["GRE Score"], kde=True, color='Darkblue')
plt.title("GRE score distribution");
sns.displot(x=df["TOEFL Score"], kde=True, color='Red')
plt.title("TOEFL score distribution");
sns.countplot(x=df["University Rating"]);
plt.title("Universities Star Rating by applicants");
```

```
sns.displot(x=df["SOP"], kde=True, color='darkgreen');
plt.title("Distribution for ratings on SOP");
sns.displot(x=df["CGPA"], kde=True, color='orange');
plt.title("Distribution of CGPA Score");
sns.countplot(x=df["Research"]);
plt.title("Research Count");
df.columns
sns.regplot(x='SOP', y='CGPA', data=df)
sns.boxplot(x='SOP',y='CGPA',data=df,color='orange')
sns.violinplot(x='SOP',y='CGPA',data=df,color='yellow')
sns.scatterplot(x='SOP', y='CGPA', data=df)
sns.pairplot(df,hue='Research')
corr_matrix = df.corr()
plt.figure(figsize = (15, 15))
sns.heatmap(corr_matrix,annot=True,fmt='0.2f')
plt.title("Correlation Matrix", fontsize = 20)
plt.show()
pd.plotting.scatter_matrix(df.loc[:,"GRE Score":"Research"],diagonal="kde",figsize=(20,15))
plt.show
x = df.iloc[:,0:7]
x
x.head()
y = df.iloc[:,7:]
y
y.head()
x.shape
y.shape
print(f'x contains: {x.shape[0]} rows and {x.shape[1]} columns')
print(f'y contains: {y.shape}')
X=df.drop(['Chance of Admit '],axis=1)
y=df['Chance of Admit ']
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.15)
x_train
x_train.shape
y_train
y_train.shape
x_test
x_test.shape
y_test
y_test.shape
from sklearn.ensemble import GradientBoostingRegressor
model = GradientBoostingRegressor()
model.fit(x_train,y_train)
model.score(x_test,y_test)
y_predict=model.predict(x_test)
```

```
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)))
y_train = (y_train>0.5)
y_test = (y_test>0.5)
y_train
y_test
from sklearn.linear_model._logistic import LogisticRegression
classifier=LogisticRegression(random_state=0)
lr=classifier.fit(x_train,y_train.ravel())
y_pred = lr.predict(x_test)
y_pred
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix

print('Accuracy Score: %f' %(accuracy_score(y_test, y_pred) * 100))
print('Recall Score: %f' %(recall_score(y_test, y_pred) * 100))
print('ROC AUC Score: %f' %(roc_auc_score(y_test, y_pred) * 100))
print('Confussion Matrix:\n', confusion_matrix(y_test, y_pred))
import pickle
pickle.dump(lr, open("university.pkl", "wb"))
pickle.dump(model, open("university_percent.pkl", "wb"))
predictions = pickle.load(open("university.pkl", "rb"))
percent = pickle.load(open("university_percent.pkl", "rb"))
from ibm_watson_machine_learning import APIClient
import json
wml_credentials = {
   "url":"https://us-south.ml.cloud.ibm.com",
   "apikey" : "M9HzyQsT95Fg-UVXiBtaNwNEl_8vdomPyIOtKwTlPohb"

}
wml_client = APIClient(wml_credentials)
wml_client.spaces.list()
SPACE_ID="1bd6f2af-83fb-4b69-b53d-ce5027f9a495"
wml_client.set.default_space(SPACE_ID)
wml_client.set.default_space(SPACE_ID)
import sklearn
sklearn.__version__
MODEL_NAME = 'Prediction'
DEPLOYMENT_NAME = 'Prediction'
DEMO_MODEL = model
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-
py3.9')
model_props = {
   wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
   wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
```

```
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=x_train,
    training_target=y_train
)
model_details
model_id = wml_client.repository.get_model_id(model_details)
model_id
deployment_props = {
    wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
    wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
deployment = wml_client.deployments.create(
    artifact_uid=model_id,
    meta_props=deployment_props
)
```

GITHUB LINK:
https://github.com/IBM-EPBL/IBM-Project-1671-1658409013

PROJECT DEMO LINK:
https://drive.google.com/file/d/1H069g2mNshNGauR_2smBFg0FA7ZPvpjb/view?usp=sharing

****