Project Report

IBM-Project-52610-1661013317

University Admit Eligibility Predictor

1. INTRODUCTION

1.1 Project Overview

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

At times they do know which stream they want to get into, but it is not easy for them to find colleges based on their academic marks and other performances.

We aim to develop and provide a place which would give a probabilistic output of how likely it is to get into a university given their details.

1.2 Purpose

- Students often feel difficulty in shortlisting the universities to apply for which they tend to wonder if their profile matches the requirements of a certain university
- The aim of this project is to help students in shortlisting universities with their profiles.
- Moreover, the cost of applying to a university is extremely high making it critical that students shortlist universities based on their profile.
- The predicted output gives them a fair idea about their admission chances to a particular university.
- This system reduces dependence on educational consultancies, who charge loads of money to analyze a candidate's profile and determine the universities he/she should apply to.

2. LITERATURE SURVEY

2.1Existing problem

- We have so many websites with problems in inaccuracy and not getting the right thing out.

- Our project, which is based on University Admit Eligibility Predictor, with a great accuracy mark, gives the output more effectively and efficiently.

2.2 References

- P.KaviPriya, “A Review on Predicting Students’ Academic Performance Earlier, Using Data Mining Techniques”, International Jour-nal of Advanced Research in Computer Science and Software Engineering
- Ali Daud, Naif Radi Aljohani, “Predicting Student Performance using Advanced Learning Analytics”, 2017 International World Wide Web Conference Committee (IW3C2).
- Marium-E-Jannat,SaymaSultana,Munira Akther, “A Probabilistic Machine Learning Approach for Eligible Candidate Selection”, Inter-national Journal of Computer Applications (0975 – 8887)Volume 144 – No.10, June 2016
- Ms. Roshani Ade,Dr. P. R. Deshmukh, “An incremental ensemble of classifiers as a technique for prediction of student’s career choice”, 2014 First International Conference on Networks & Soft Compu-ting
- Nikita Gorad ,Ishani Zalte, “Career Counselling Using Data Mining”, International Journal of Innovative Research in Computer and Communication Engineering.
- Bo Guo , Rui Zhang, “Predicting Students Performance in Educa-tional Data Mining”,2015 International Symposium on Educational Technology
- Ali Daud , Naif Radi Aljohani , “Predicting Student Performance using Advanced Learning Analytics”
- Rutvija Pandya Jayati Pandya , “C5.0 Algorithm to Improved KNN with Feature Selection and Reduced Error Pruning“, Inter-national Journal of Computer Applications (0975 – 8887) Volume 117 – No. 16, May 2015.
- Comparative Analysis of KNN Algorithms: ID3, C4.5 and KNNShiju Sathyadevan and Remya R. Nair
- Yu Lou, Ran Ren, “A Machine Learning Approach for Future Ca-reer Planning”
- Gareth James ,Daniela Witten ,Trevor Hastie, ”An Introduction to Statistical Learning with Applications in R”
- Anuj Karpatne, Gowtham Atluri, “Theory- Guided Data Science: A New Paradigm for Scientific Discovery from Data”, IEEE trans-actions on knowledge and data engineering, vol.29, no. 10, october 2017.

2.3 Problem Statement Definition

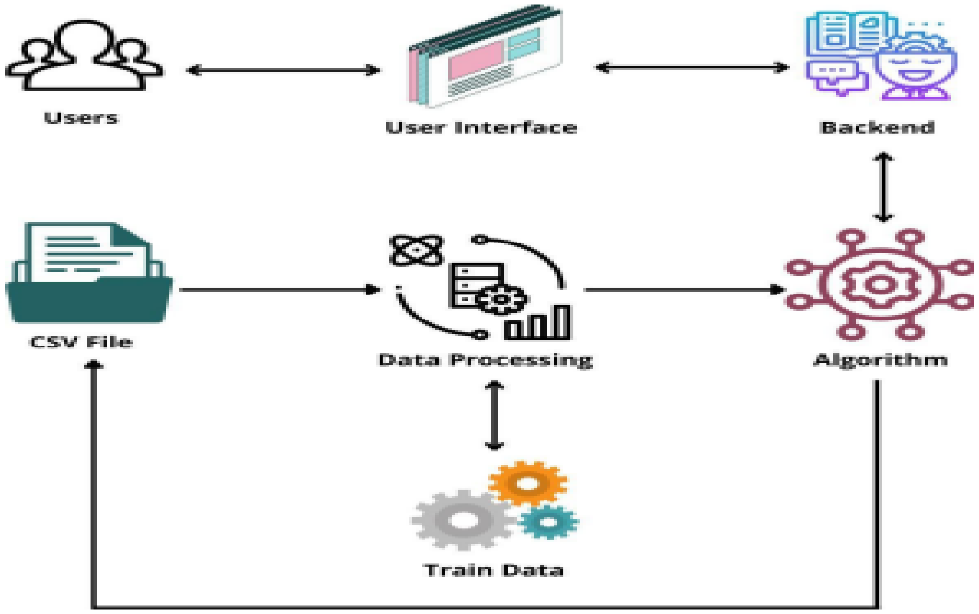
Problem Statement

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

- At times they do know which stream they want to get into, but it is not easy for them to find colleges based on their academic marks and other performances.
- We aim to develop and provide a place which would give a probabilistic output of how likely it is to get into a university given their details.

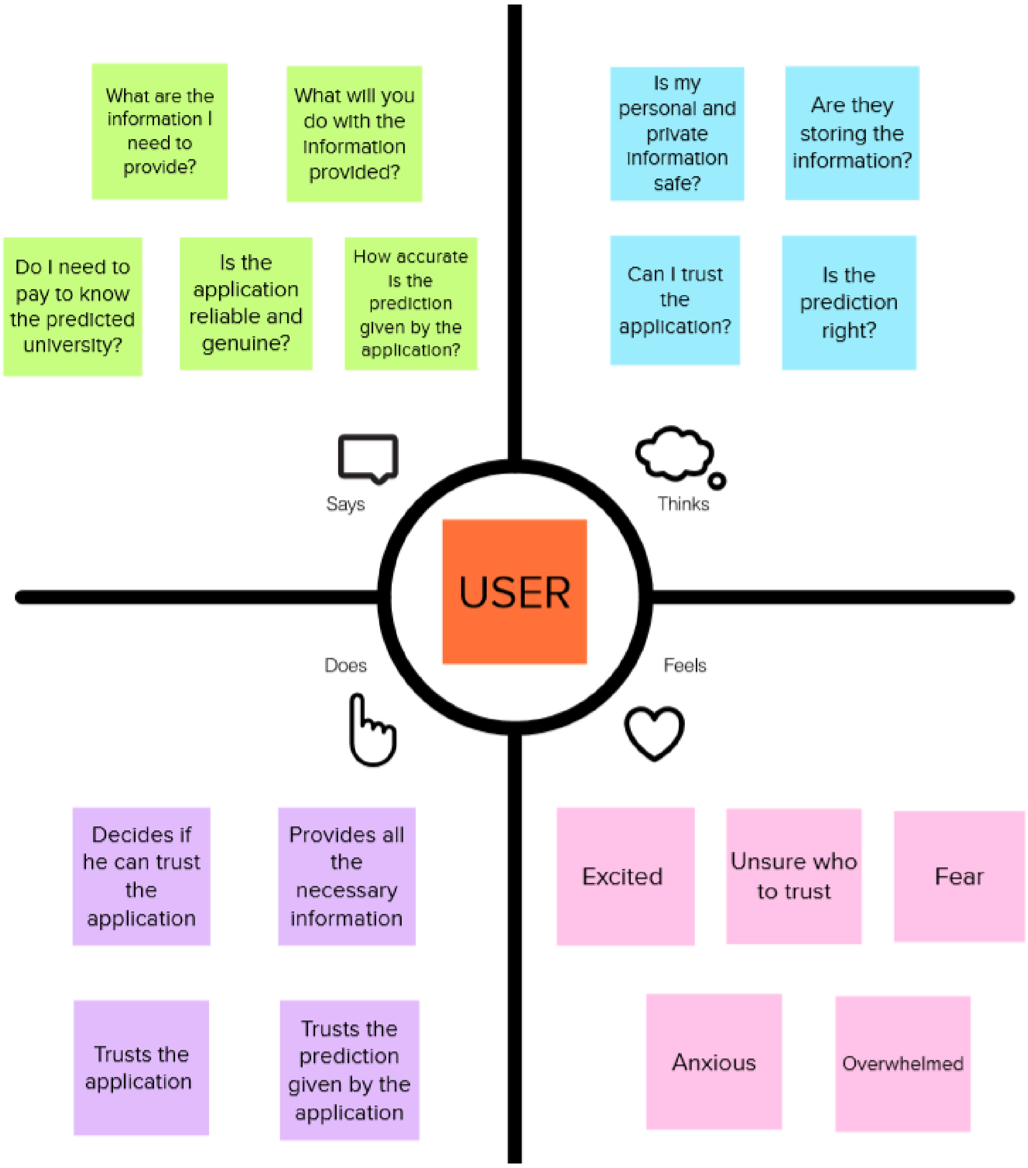
Abstract:

- 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 to a particular university.
- This analysis should also help students who are currently preparing or will be preparing to get a better idea.

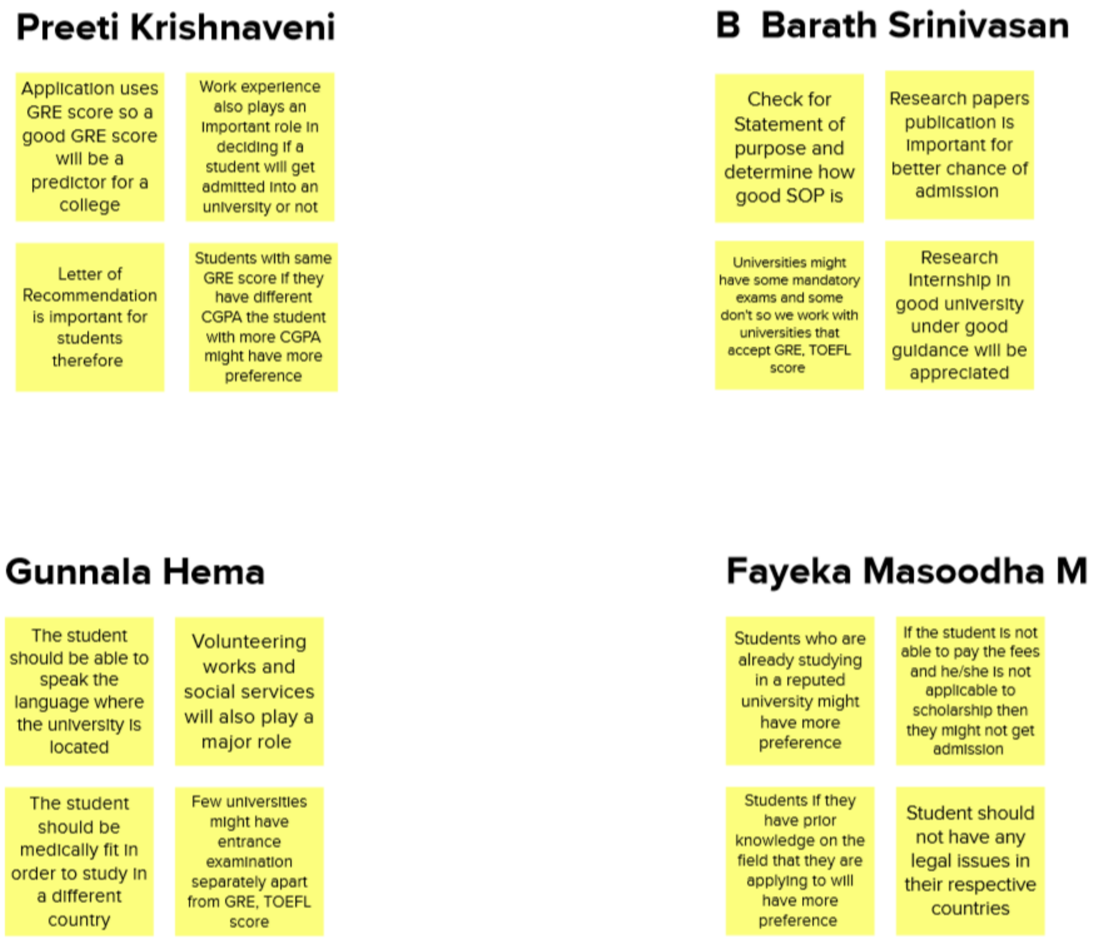


3. IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas


What are the information I need to provide?
What will you do with the information provided?
Do I need to pay to know the predicted university?
Is the application reliable and genuine?
How accurate is the prediction given by the application?
Is my personal and private information safe?
Are they storing the information?
Can I trust the application?
Is the prediction right?
Says
Thinks
USER
Does
Feels
Decides if he can trust the application
Provides all the necessary information
Trusts the application
Trusts the prediction given by the application
Excited
Unsure who to trust
Fear
Anxious
Overwhelmed

3.2Ideation & Brainstorming



Preeti Krishnaveni

Application uses GRE score so a good GRE score will be a predictor for a college

Work experience also plays an important role in deciding if a student will get admitted into an university or not

Letter of Recommendation is important for students therefore

Students with same GRE score if they have different CGPA the student with more CGPA might have more preference

B Barath Srinivasan

Check for Statement of purpose and determine how good SOP is

Research papers publication is important for better chance of admission

Universities might have some mandatory exams and some don't so we work with universities that accept GRE, TOEFL score

Research Internship in good university under good guidance will be appreciated

Gunnala Hema

The student should be able to speak the language where the university is located

Volunteering works and social services will also play a major role

The student should be medically fit in order to study in a different country

Few universities might have entrance examination separately apart from GRE, TOEFL score

Fayeka Masoodha M

Students who are already studying in a reputed university might have more preference

If the student is not able to pay the fees and he/she is not applicable to scholarship then they might not get admission

Students if they have prior knowledge on the field that they are applying to will have more preference

Student should not have any legal issues in their respective countries

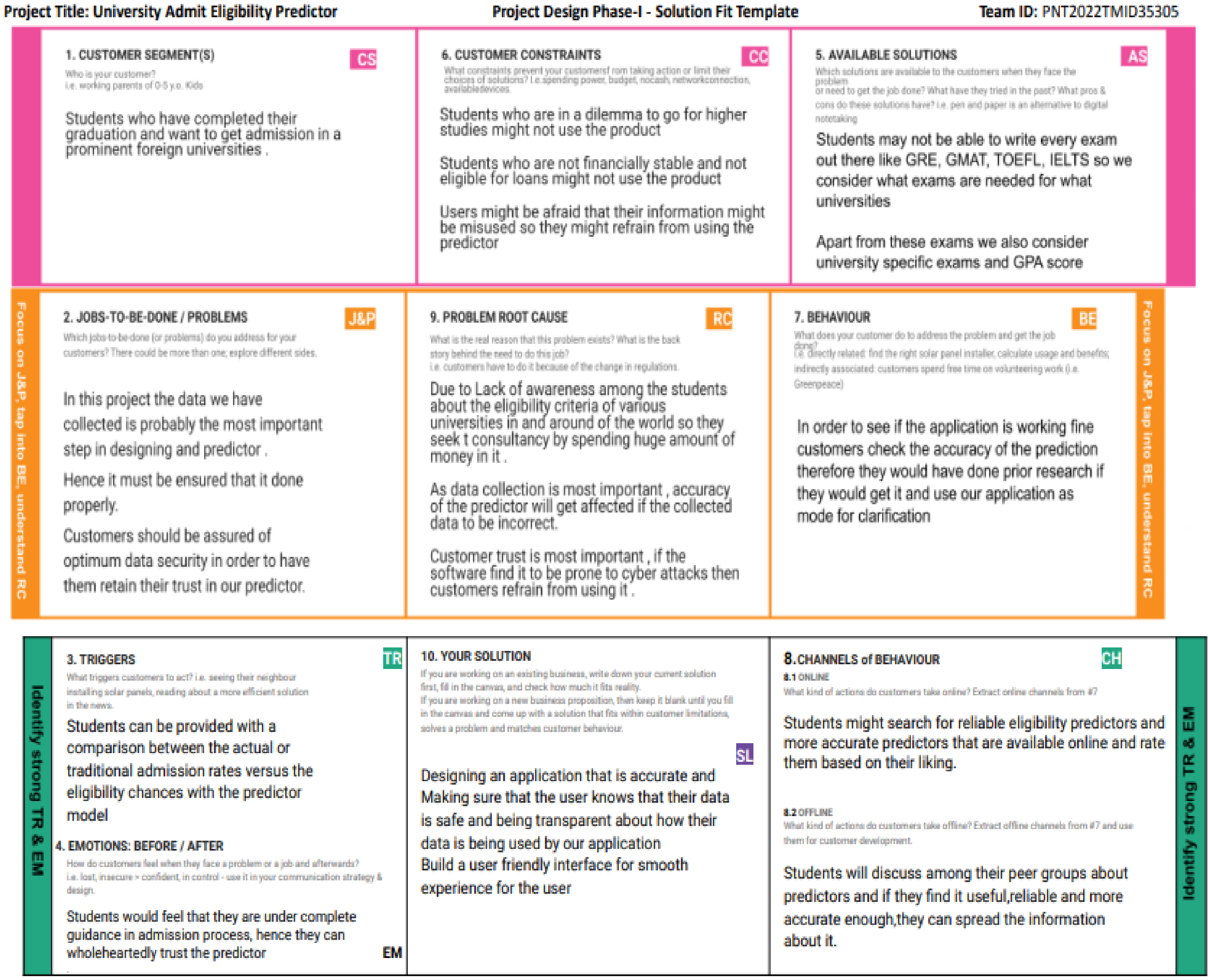
3.3 Proposed Solution

Proposed Solution :

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	• Students are often worried about their chances of admission to university. • Students have to go college by college to get admission, which takes a lot of time and money. • When students come from rural places, they find it hard to go along with the formal procedures. • Stressful conditions may occur while searching for best university
2	Idea / Solution description	• University and College research being one part of the university application process is itself an arduous and lengthy task. This issue being a big problem for students has not been solved till now. • There are recognized sites which

		filter the best universities and colleges based on the location, tuition fees, major and degree but none of them have used machine learning algorithms to solve the issue. ● Hence, we have done this research project to solve that issue to some extent with the use of machine learning techniques.
3	Novelty / Uniqueness	● Most of the available project used only GRE score but our project considers all the factors like SOP, LOR, Research projects for admission. ● Our project has a working GUI that makes it easier for students to use our product effectively.
4	Social Impact / Customer Satisfaction	● It reduces the ambiguity among students about college admission. ● It gives a clear idea about the requirements for the admission process in any university.
5	Business Model (Revenue Model)	● This predictor system can be used by consultancy firms to help their customers who seek their help. Apart from this valuable suggestion can be provided for an appropriate subscription fee
6	Scalability of the Solution	● Training the model with even more attributes will increases the efficiency further. ● The solution could be scaled up to include more universities and larger geographical area.

3.4Problem Solution fit



Project Title: University Admit Eligibility Predictor **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID35305

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. Kids

Students who have completed their graduation and want to get admission in a prominent foreign universities .

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customersf rom taking action or limit their choices of solutions? i.e.spending power, budget, nocash, networkconnection, availabledevices.

Students who are in a dilemma to go for higher studies might not use the product

Students who are not financially stable and not eligible for loans might not use the product

Users might be afraid that their information might be misused so they might refrain from using the predictor

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Students may not be able to write every exam out there like GRE, GMAT, TOEFL, IELTS so we consider what exams are needed for what universities

Apart from these exams we also consider university specific exams and GPA score

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

In this project the data we have collected is probably the most important step in designing and predictor .

Hence it must be ensured that it done properly.

Customers should be assured of optimum data security in order to have them retain their trust in our predictor.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Due to Lack of awareness among the students about the eligibility criteria of various universities in and around of the world so they seek t consultancy by spending huge amount of money in it .

As data collection is most important , accuracy of the predictor will get affected if the collected data to be incorrect.

Customer trust is most important , if the software find it to be prone to cyber attacks then customers refrain from using it .

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

In order to see if the application is working fine customers check the accuracy of the prediction therefore they would have done prior research if they would get it and use our application as mode for clarification

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Students can be provided with a comparison between the actual or traditional admission rates versus the eligibility chances with the predictor model

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Students would feel that they are under complete guidance in admission process, hence they can wholeheartedly trust the predictor EM

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Designing an application that is accurate and Making sure that the user knows that their data is safe and being transparent about how their data is being used by our application
Build a user friendly interface for smooth experience for the user

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

Students might search for reliable eligibility predictors and more accurate predictors that are available online and rate them based on their liking.

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

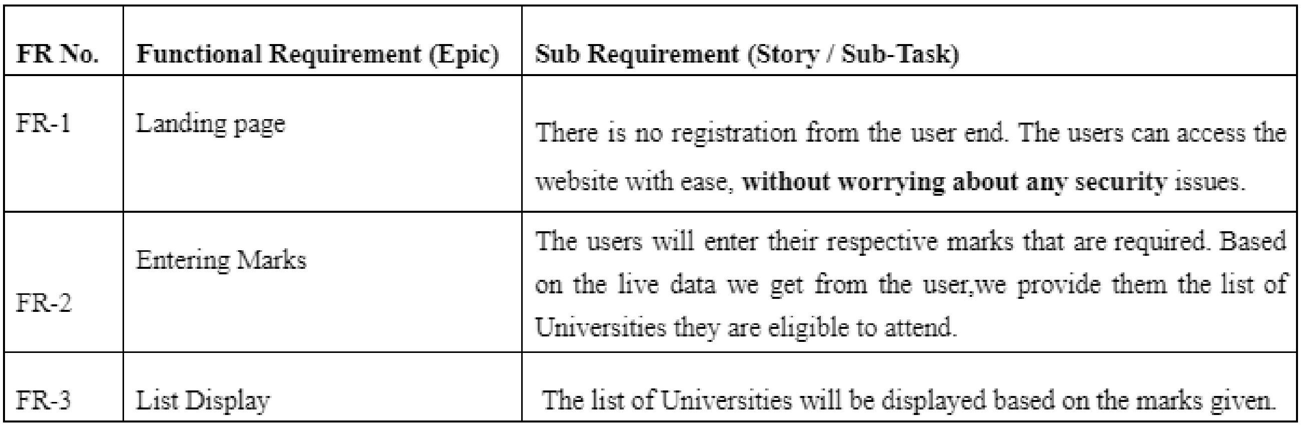
Students will discuss among their peer groups about predictors and if they find it useful,reliable and more accurate enough,they can spread the information about it.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1Functional requirement

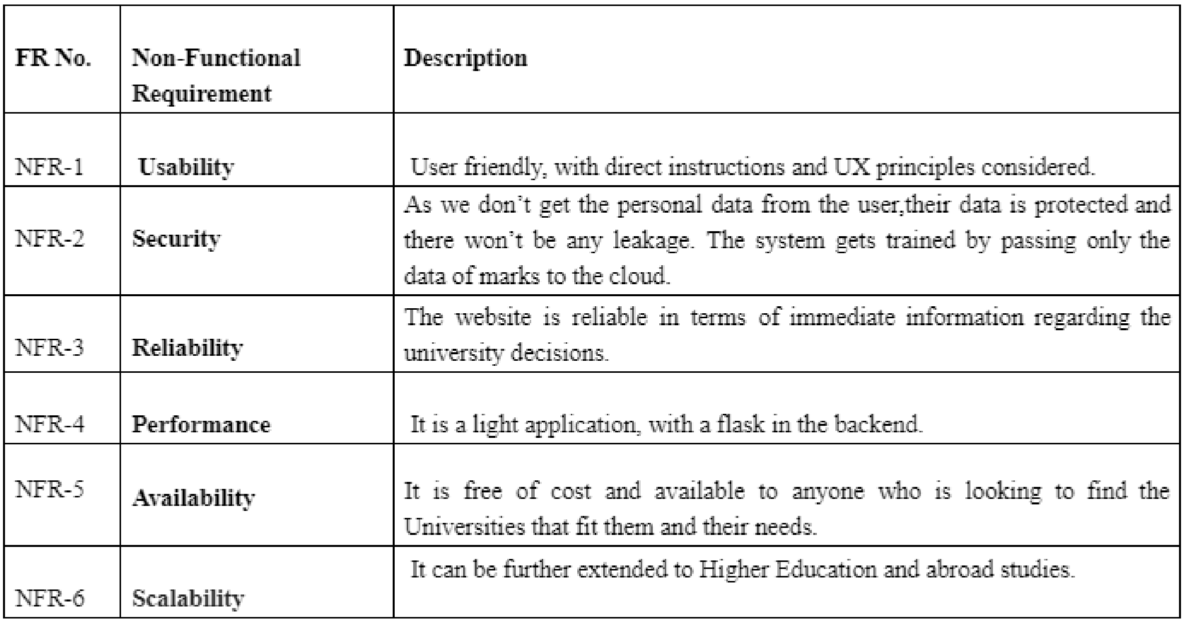
Following are the functional requirements of the proposed solution.



FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Landing page	There is no registration from the user end. The users can access the website with ease, **without worrying about any security** issues.
FR-2	Entering Marks	The users will enter their respective marks that are required. Based on the live data we get from the user,we provide them the list of Universities they are eligible to attend.
FR-3	List Display	The list of Universities will be displayed based on the marks given.

4.2 Non-Functional requirements

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

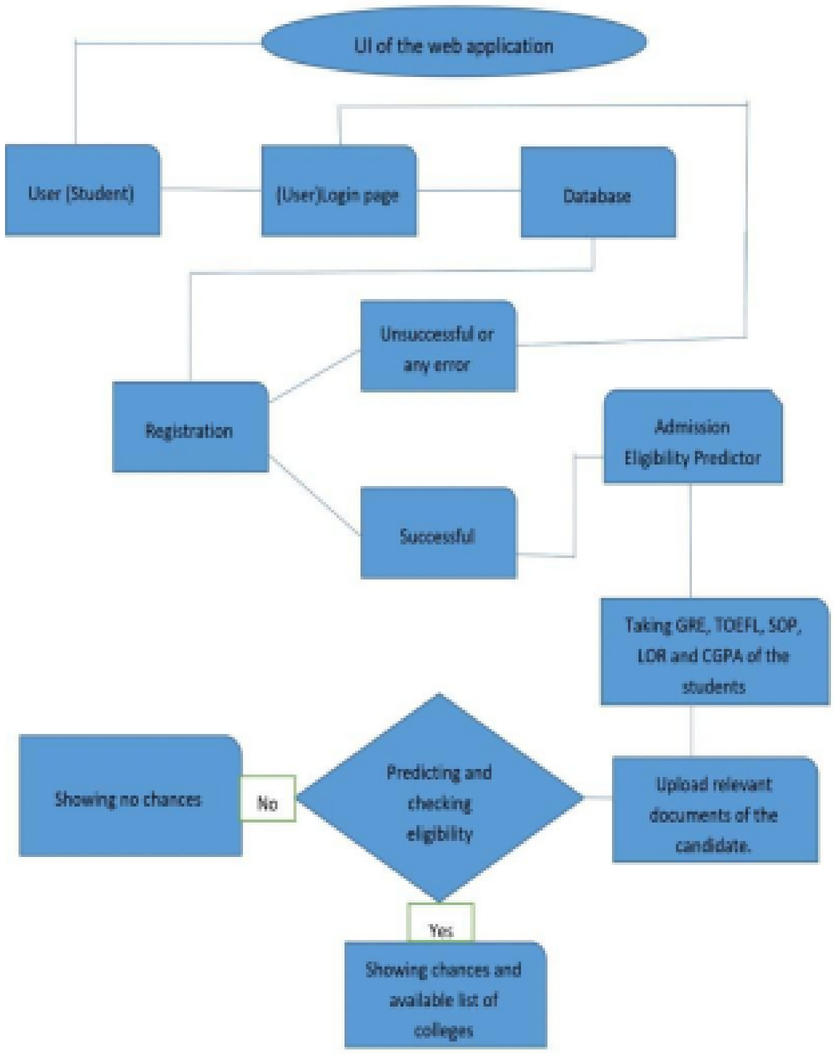


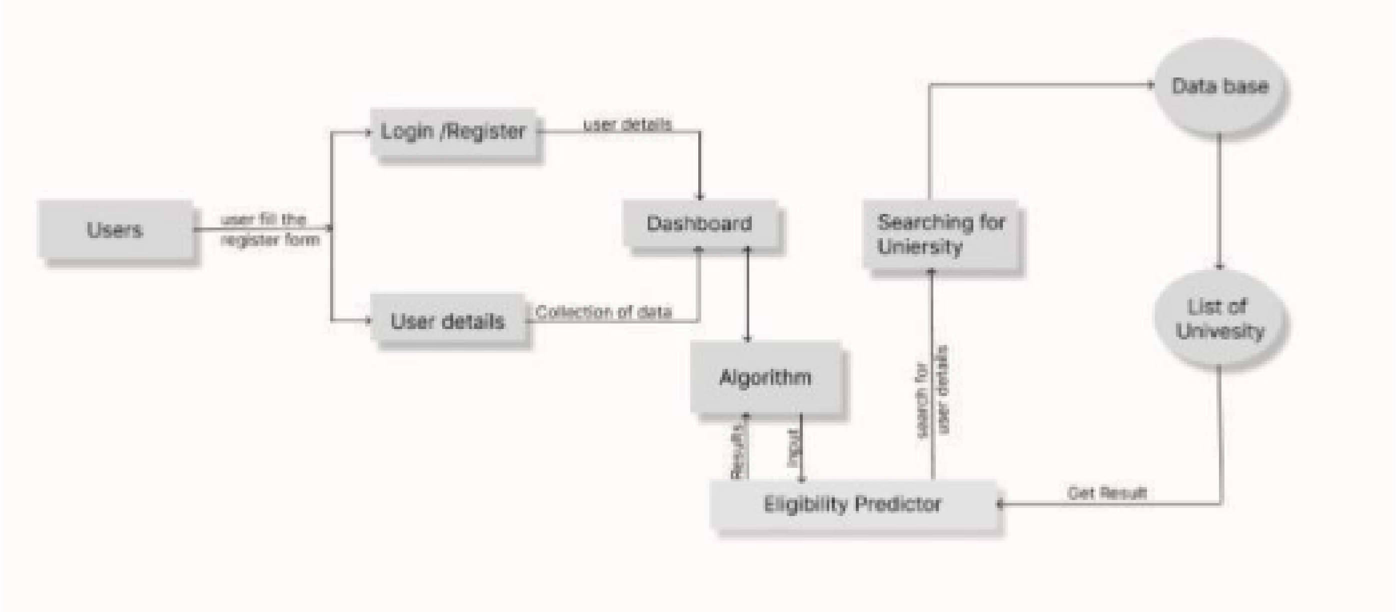
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User friendly, with direct instructions and UX principles considered.
NFR-2	**Security**	As we don't get the personal data from the user,their data is protected and there won't be any leakage. The system gets trained by passing only the data of marks to the cloud.
NFR-3	**Reliability**	The website is reliable in terms of immediate information regarding the university decisions.
NFR-4	**Performance**	It is a light application, with a flask in the backend.
NFR-5	**Availability**	It is free of cost and available to anyone who is looking to find the Universities that fit them and their needs.
NFR-6	**Scalability**	It can be further extended to Higher Education and abroad studies.

5. PROJECT DESIGN

5.1Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


UI of the web application
User (Student)
(User)Login page
Database
Unsuccessful or any error
Registration
Admission Eligibility Predictor
Successful
Taking GRE, TOEFL, SOP, LOR and CGPA of the students
Showing no chances
No
Predicting and checking eligibility
Upload relevant documents of the candidate.
Yes
Showing chances and available list of colleges

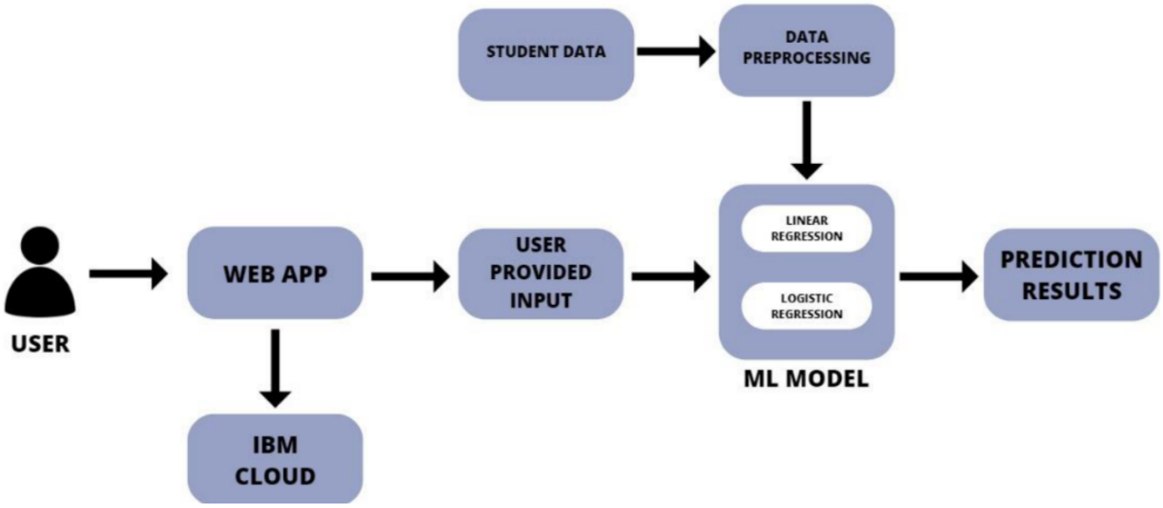


User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer 1 (Web user)	Landing Page	USN-1	It is pretty clear about why we use this website and how it should be used, just by looking at this page.	Understandable	High	Sprint-1
Customer 2 (Web User)	Landing Page	USN-2	The concept of the application is clear with all the how to do instructions and everything.	Very clear	High	Sprint-1
Customer 3 (Web User)	Marks page	USN-3	Its pretty clear what kind of data should be given in.	I can give my marks details.	Low	Sprint-2
Customer 4 (Web User)	Results Page	USN-4	I can see the right and correct results based on the marks given to the system	Got the details	High	Sprint-1

5.2Solution & Technical Architecture

Architecture Diagram:



Solution Architecture :

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 to a particular university.

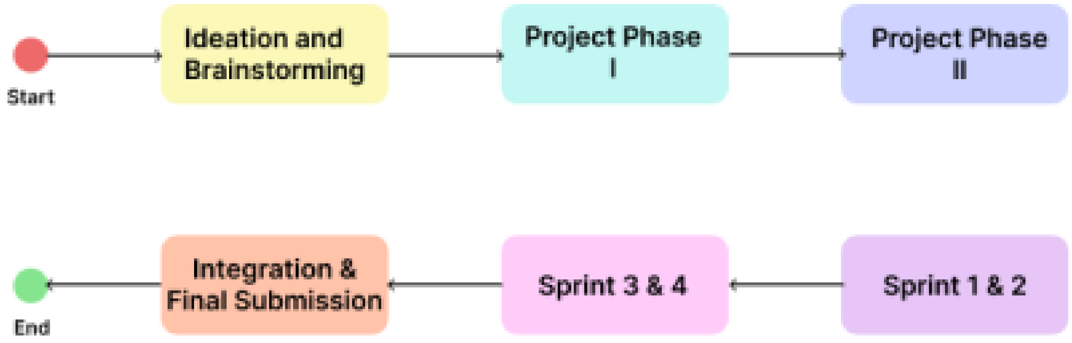
This analysis should also help students who are currently preparing or will be preparing to get a better idea

5.3User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer 1 (Web user)	Landing Page	USN-1	It is pretty clear about why we use this website and how it should be used, just by looking at this page.	Understandable	High	Sprint-1
Customer 2 (Web User)	Landing Page	USN-2	The concept of the application is clear with all the how to do instructions and everything.	Very clear	High	Sprint-1
Customer 3 (Web User)	Marks page	USN-3	Its pretty clear what kind of data should be given in.	I can give my marks details.	Low	Sprint-2
Customer 4 (Web User)	Results Page	USN-4	I can see the right and correct results based on the marks given to the system	Got the details	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



- The sprints started right after our Training session by IBM.

- Though it started then, we started brainstorming our project since the beginning of this semester's calendar.

- We completed the “Ideation and Brainstorming” phase first, moving on to each other phases one by one.

- Each took a task (exactly what we estimated) to complete.

- First Sprint contains the HTML Code, which acts as a building block for our application.

- Second Sprint is the CSS Code, which enhances the look of the website.

- Third Sprint is the initialization of the flask language and a little bit of backend code. This is where we learned all the important topics needed to complete this project.

- Then the last Sprint, the fourth one, contains only the backend python-flask code, that performs various Data Manipulation and trains the model

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Frontend - HTML	USN-1	I'm able to see the tables or columns where I can inject my marks into.	2	Medium	Gunnala Hema
Sprint-2	Frontend - CSS	USN-2	Now the application looks more appealing and nice to the eyes.	1	Low	Fayeka Masoodha M
Sprint-3	Flask	USN-3	I can see that my data is being processed.	2	Medium	Preeti Krishnaveni Ra
Sprint-4	Python	USN-4	I can get the results from the inputs I have given to the system.	2	High	B Barath Srinivasan

Sprint	Total Story Points	Duration	Sprint Start date	Sprint End Date	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022 20	17.11.22
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022 20	17.11.22
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022 20	17.11.22
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022 20	17.11.22

7. CODING & SOLUTIONING

7.1 Feature 1

HTML CODE:

```
<!DOCTYPE html>


<html lang="en">

<head>

   <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

   <title>Login</title>

   <link rel="stylesheet" href="login-style.css">

   <link href="https://fonts.googleapis.com/css?family=Raleway:700,500,1000&display=swap"
rel="stylesheet">

   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js">
</script>


<script>

   $(function() {




  var people = [];

  $.getJSON('  https://api.thingspeak.com/channels/1013258/feeds.json?results=1', function(data) {

     $.each(data.feeds, function(i, f) {

       var tblRow = "<tr>" + "<td>" + f.created_at + "</td>" +"<td>" + f.entry_id + "</td>" + "<td>"
+ f.field8 + "</td>"+"</tr>"

        $(tblRow).appendTo("#userdata tbody");

    });

  });
});

</script>

</head>

<body>

  <main>

    <div class="background">

      <div class="text">

        <center style="color:blue">

        <h1>PREDICTING COLLEGES</h1>

</center><br>

<center>

<form  action="https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&"
method="post" target="_blank">

 <div class="form-group">

  <label>GRE Score</label>

  <input type="text" class="form-control" name="field1">

 </div>

 <div class="form-group">

  <label>TOEFL Score</label>

  <input type="text" class="form-control" name="field2">

 </div>

 <div class="form-group">

  <label>UNVERSITY RATING</label>

  <input type="text" class="form-control" name="field3">

 </div>

 <div class="form-group">

  <label>SOP</label>

  <input type="text" class="form-control" name="field4">

 </div>

 <div class="form-group">

  <label>LOR</label>
```

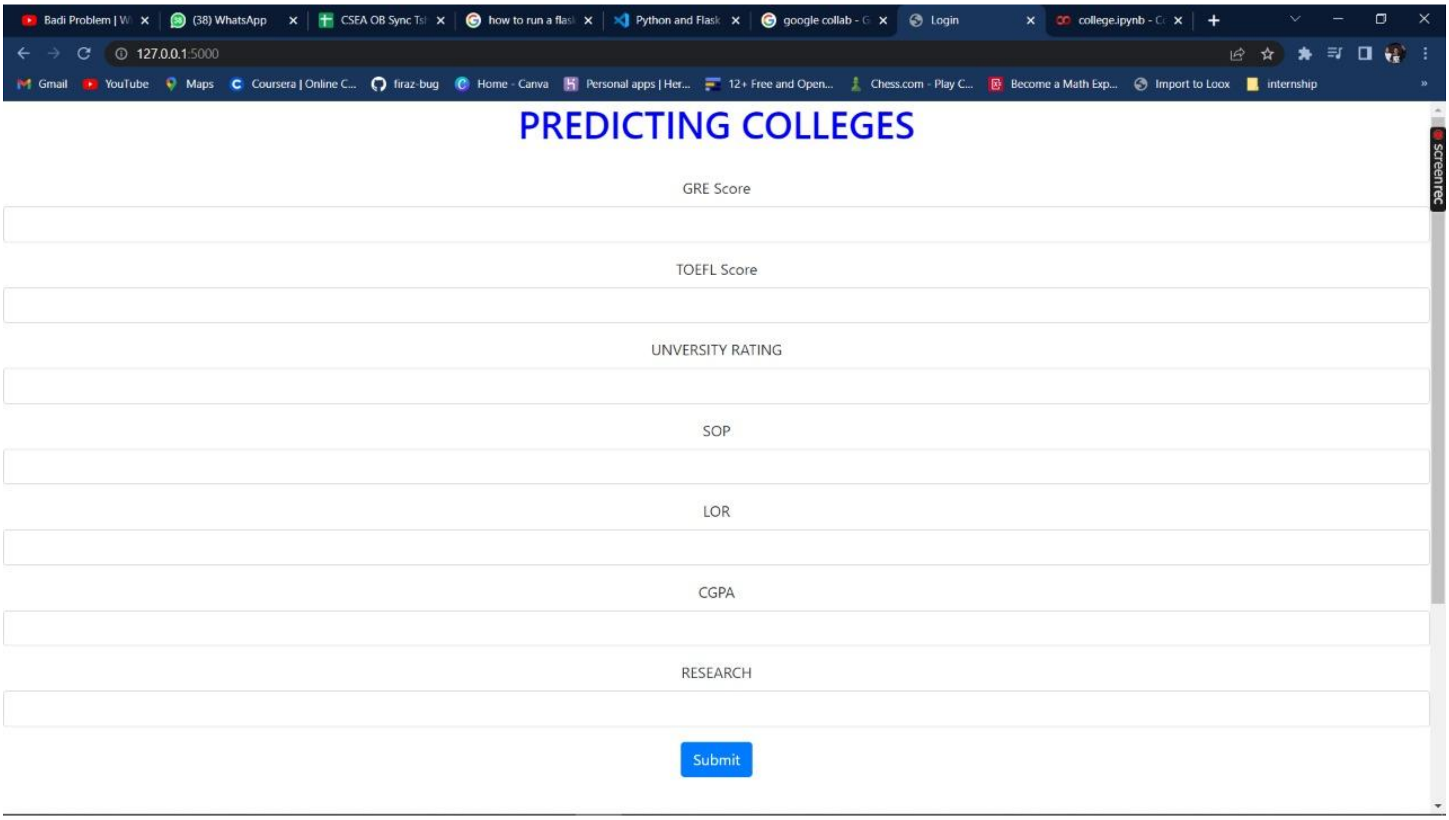
```
    <input type="text" class="form-control" name="field5">
  </div>
  <div class="form-group">
    <label>CGPA</label>
    <input type="text" class="form-control" name="field6">
  </div>
  <div class="form-group">
    <label>RESEARCH</label>
    <input type="text" class="form-control" name="field7">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<table id= "userdata" border="2">
  <thead>
            <th>Date</th>
            <th>S.no</th>
            <th>COLLAGES</th>
             </thead>
      <tbody>
       </tbody>
   </table>
```

```
<iframe id='track' frameborder="0" scrolling="no" width="1" height="1">

</script>

</center>

          </div>

       </div>

    </main>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1
" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

</body>

</html>
```

CSS CODE:

```
 {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    text-decoration: none;

}

body{

    font-family: 'Raleway', sans-serif;

    background: #000;

}

.background{
    background: url(background.jpeg) no-repeat;
  background-position: center top;
background-size: contain;
 height:2200px;
 position:relative
  display: flex;
 }
 .text, .box{
    margin-top:0vh;
    flex: 1;
 }
 .text{
    margin-left: 0%;
    font-weight: 200px;
 color:white;
 }
 .box{
    margin-right: 25%;
 }
 .text h1{
    font-size: 70px;
    color: #fff;
    font-weight: 500;
 }
 .text h2{
    font-size: 70px;
    color: #fff;
    font-weight: 500;
```

```
}
.text p{
   font-size: 20px;
   color: #fff;
   font-weight: 300;
}
.text p a{
   color: #fff;
   font-weight: 700;
}
.form{
   background: transparent;
   color: #fff;
   box-sizing: border-box;
   display: flex;
   flex-direction: column;
   width:50%;
}
input{
   margin: 20px 0;
   padding: 10px;
   background: transparent;
   border: none;
   outline: none;
   color: #fff;
   font-family: 'Raleway', sans-serif;
}
.username, .password{
  border-bottom: 1px solid #fff;
}
.button{
  background: transparent;
  border: 1px solid #fff;
  color: #fff;
  font-size: 18px;
}
.button:hover{
  background: #000;
  color: #fff;
}
```



PREDICTING COLLEGES

GRE Score

TOEFL Score

UNVERSITY RATING

SOP

LOR

CGPA

RESEARCH

Submit

7.2 Feature 2

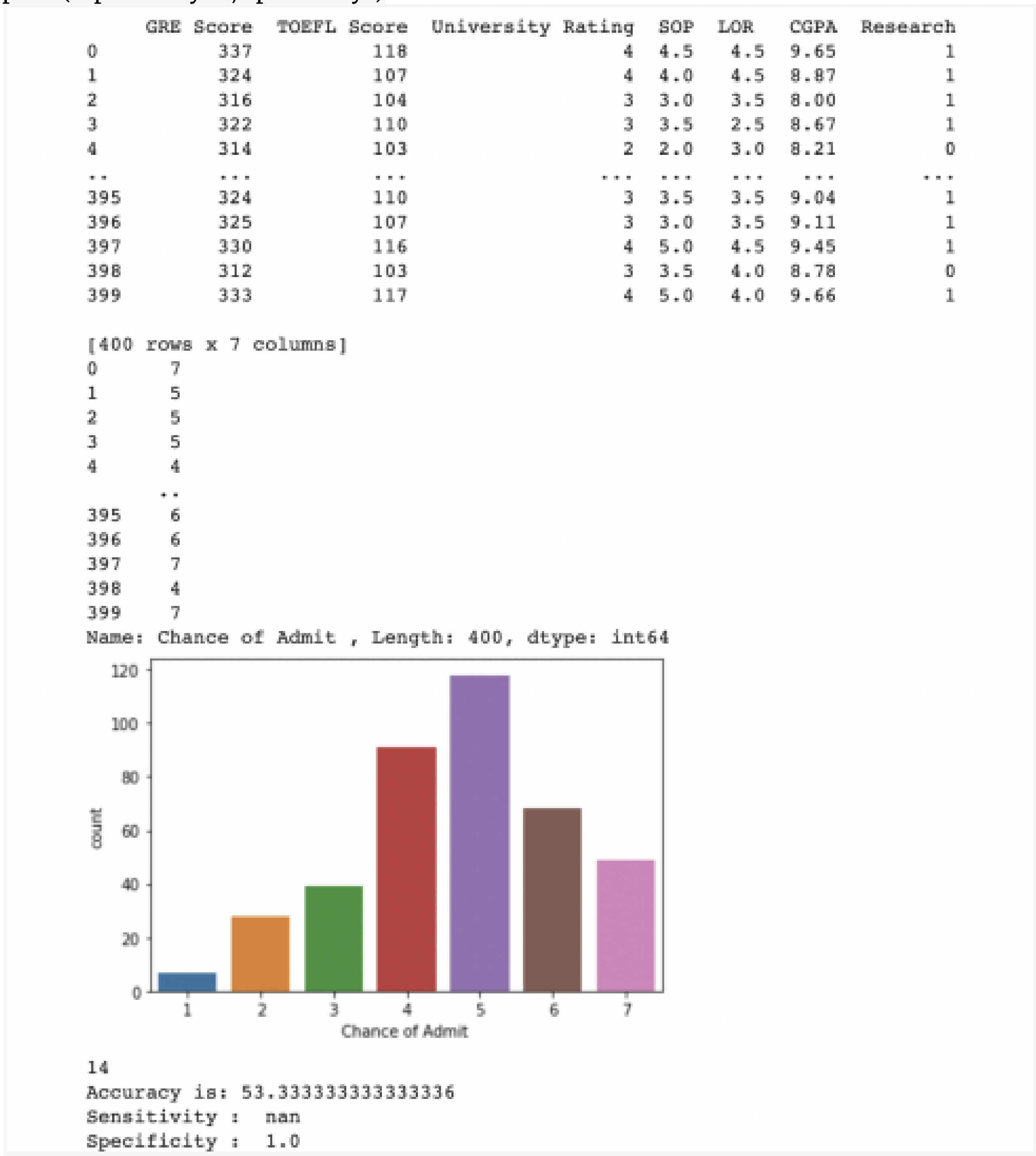
college.ipynb

```
import numpy as np
import pandas as pd
```

```
from sklearn import metrics

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt
import seaborn as sns

import pickle

data = pd.read_csv('data.csv')

data.head()

data.shape

X = data.iloc[:,:-1]

X.head()

y = data.iloc[:,-1]

y.head()

print(X)

print(y)

data['Chance of Admit '].value_counts()

X_train,X_test,y_train,y_test =

train_test_split(X,y,test_size=0.3,random_state=1)

sns.countplot(x='Chance of Admit ',data=data)

plt.show()

X_train.shape

X_train.head()

y_test.shape

y_test.head()

from sklearn.metrics import accuracy_score

max_accuracy = 0


from sklearn.neighbors import KNeighborsClassifier
```

```
for x in range(1,100):

    model = KNeighborsClassifier(n__neighbors=x)

    model.fit(X__train,y__train)
    y__pred = model.predict(X__test)

    current__accuracy =

    round(accuracy__score(y__pred,y__test)*100,2)

    if(current__accuracy>max__accuracy):

        max__accuracy = current__accuracy

        best__x = x


#print(max_accuracy)

print(best__x)


model =

KNeighborsClassifier(n__neighbors=best

__x) model.fit(X__train,y__train)

y__pred = model.predict(X__test)

filename = 'knn.sav'

pickle.dump(model, open(filename, 'wb'))

acc=(metrics.accuracy__score(y__pred,y__test)*100)

print("Accuracy is:",acc)

cm1 = metrics.confusion__matrix(y__pred,y__test)


total1=sum(sum(cm1))




sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
```

```
print('Sensitivity : ', sensitivity1 )


specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])

print('Specificity : ', specificity1)
```

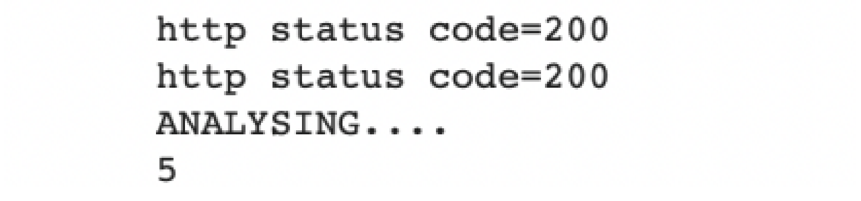


```
     GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  Research
0          337          118                  4  4.5   4.5  9.65         1
1          324          107                  4  4.0   4.5  8.87         1
2          316          104                  3  3.0   3.5  8.00         1
3          322          110                  3  3.5   2.5  8.67         1
4          314          103                  2  2.0   3.0  8.21         0
..         ...          ...                ...  ...   ...   ...       ...
395        324          110                  3  3.5   3.5  9.04         1
396        325          107                  3  3.0   3.5  9.11         1
397        330          116                  4  5.0   4.5  9.45         1
398        312          103                  3  3.5   4.0  8.78         0
399        333          117                  4  5.0   4.0  9.66         1

[400 rows x 7 columns]
0      7
1      5
2      5
3      5
4      4
      ..
395    6
396    6
397    7
398    4
399    7
Name: Chance of Admit , Length: 400, dtype: int64
```

```
14
Accuracy is: 53.333333333333336
Sensitivity :  nan
Specificity :  1.0
```

```
import pickle
import urllib.request
import json
from time import sleep
while True:
    conn =
urllib.request.urlopen("https://api.thingspeak.com/channels/1013258/feeds.json?results=1")
```

```
    response = conn.read()
    print ("http status code=%s" % (conn.getcode()))
    data=json.loads(response)
    x=int(data['feeds'][0]['entry_id'])
    y=x
    conn.close()
    while x==y:
        conn = urllib.request.urlopen("https://api.thingspeak.com/channels/1013258/feeds.json?results=1")
        response = conn.read()
        #print ("http status code=%s" % (conn.getcode()))
        data=json.loads(response)
        y=int(data['feeds'][0]['entry_id'])
        conn.close()

    conn = urllib.request.urlopen("https://api.thingspeak.com/channels/1013258/feeds.json?results=1")
    response = conn.read()
    print ("http status code=%s" % (conn.getcode()))
    data=json.loads(response)
    a=float(data['feeds'][0]['field1'])
    b=float(data['feeds'][0]['field2'])
    c=float(data['feeds'][0]['field3'])
    d=float(data['feeds'][0]['field4'])
    e=float(data['feeds'][0]['field5'])
    f=float(data['feeds'][0]['field6'])
    g=float(data['feeds'][0]['field7'])

    conn.close()
    filename = 'knn.sav'
    loaded_model = pickle.load(open(filename, 'rb'))
    person_reports = [[a,b,c,d,e,f,g]]
    predicted = loaded_model.predict(person_reports)
    print("ANALYSING....")
    print(predicted[0])
    sleep(15)
    if predicted[0]==1:
        conn = urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL& field8=1-VIT
2-JPR
3-AGNI")
    elif predicted[0]==2:
        conn = urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL& field8=1-SREC
2-KEC
3-KPR")
   elif predicted[0]==3:
         conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OE
IL& field8=1-KONGU
2-KCT
3-HIT")
   elif predicted[0]==4:
         conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OE
IL& field8=1-SASTHRA
2-SKCET
3-BIT")
   elif predicted[0]==5:
         conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OE
IL& field8=1-SRM
2-THIAGARAJAR
3-NIIT")
   elif predicted[0]==6:
         conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OE
IL& field8=1-PSG
2-CIT
3-GCT")
   elif predicted[0]==7:
         conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OE
IL& field8=1-IIT
2-MIT
3-ANNA_UNIVERSITY-CHE")
```



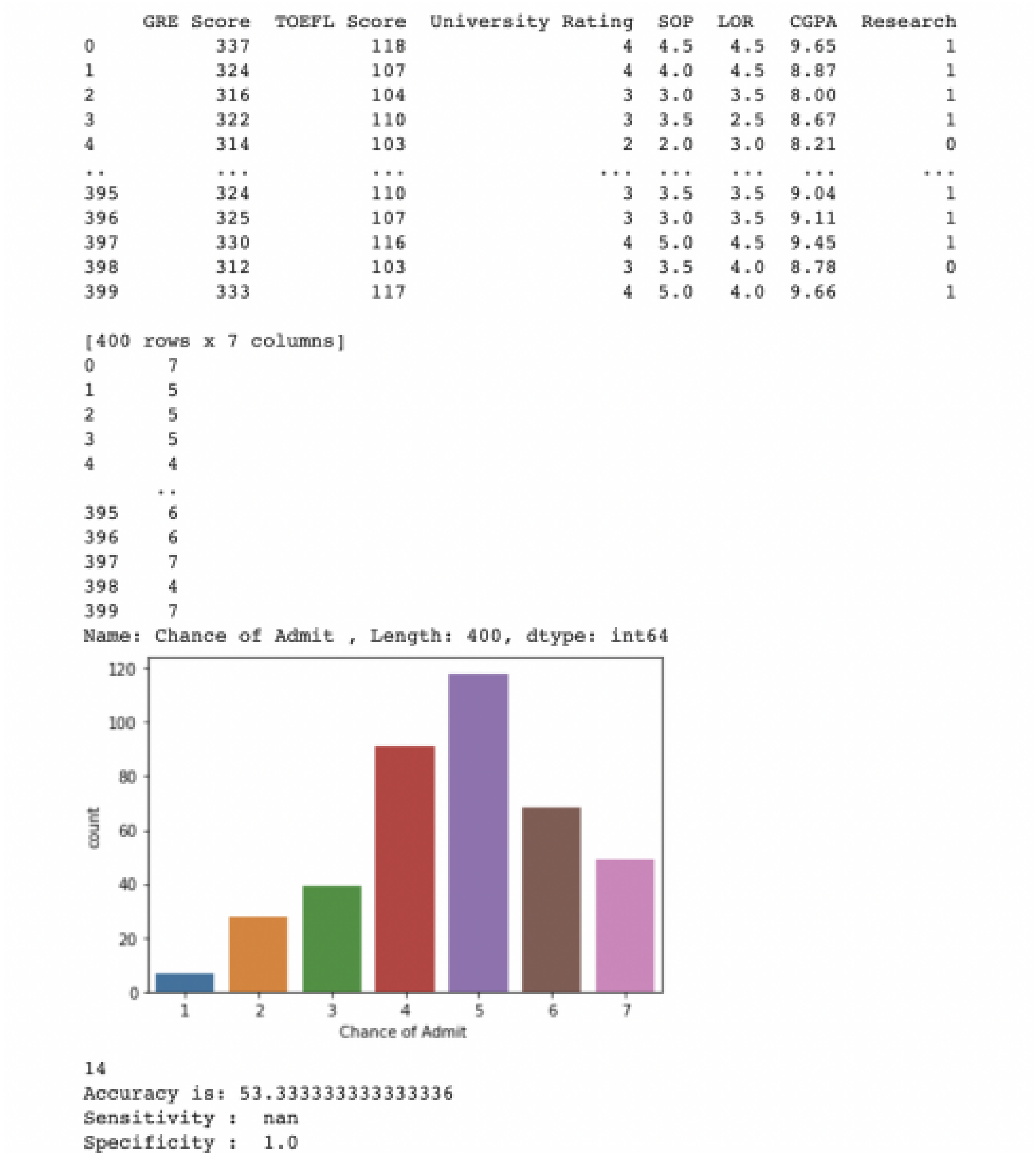
```
http status code=200
http status code=200
ANALYSING....
5
```

Integration:

```
from flask import Flask, render_template
app = Flask(__name__)
import os
import subprocess
@app.route(r'/')
def index():
    return render_template('login.html')
if __name__ == '__main__':
    app.run(debug=True)
```

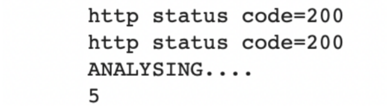
8. TESTING

8.1 Test Cases



```
     GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  Research
0          337          118                  4  4.5   4.5  9.65         1
1          324          107                  4  4.0   4.5  8.87         1
2          316          104                  3  3.0   3.5  8.00         1
3          322          110                  3  3.5   2.5  8.67         1
4          314          103                  2  2.0   3.0  8.21         0
..         ...          ...                ...  ...   ...   ...       ...
395        324          110                  3  3.5   3.5  9.04         1
396        325          107                  3  3.0   3.5  9.11         1
397        330          116                  4  5.0   4.5  9.45         1
398        312          103                  3  3.5   4.0  8.78         0
399        333          117                  4  5.0   4.0  9.66         1

[400 rows x 7 columns]
0      7
1      5
2      5
3      5
4      4
      ..
395    6
396    6
397    7
398    4
399    7
Name: Chance of Admit , Length: 400, dtype: int64
```

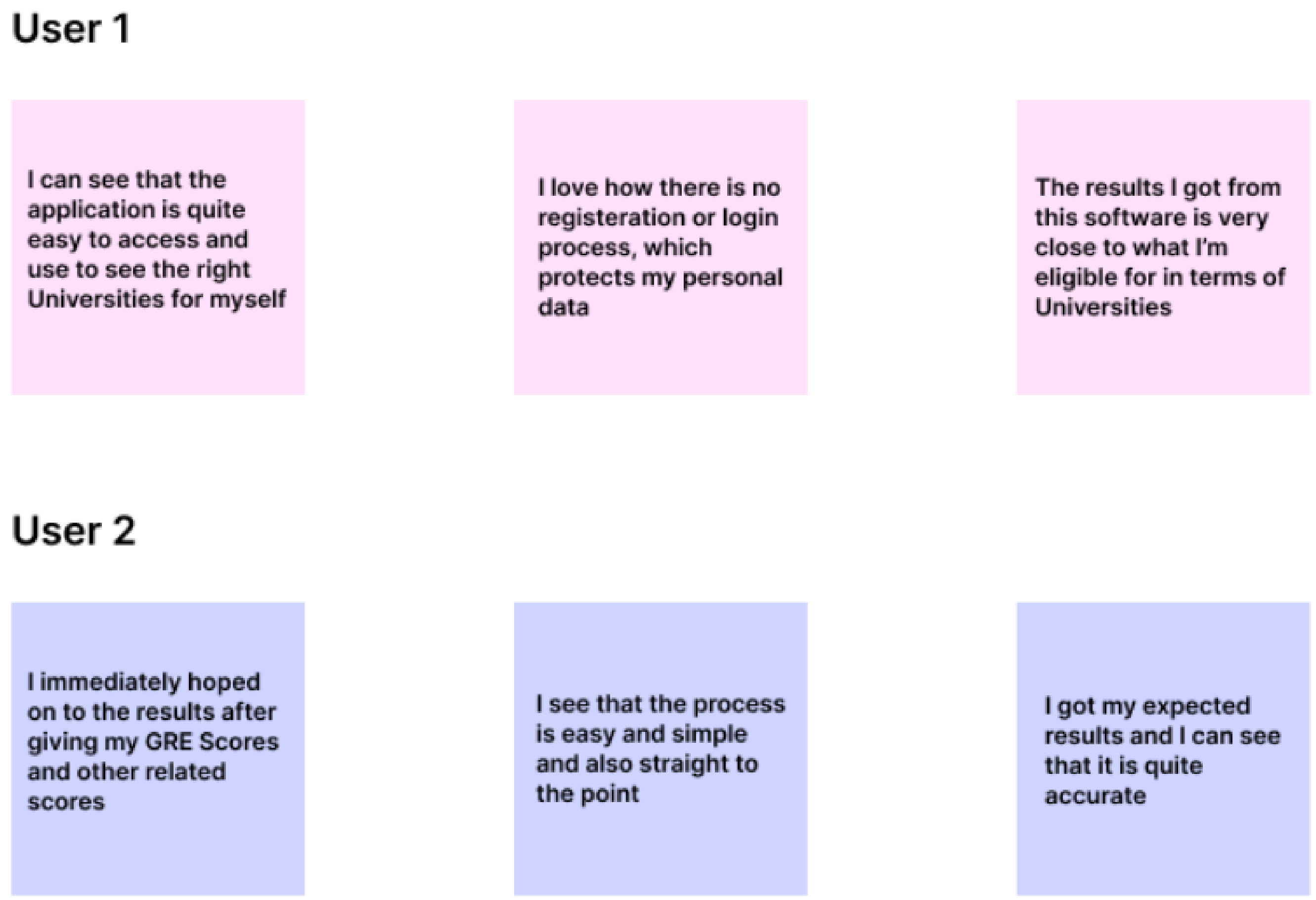
```
14
Accuracy is: 53.333333333333336
Sensitivity :  nan
Specificity :  1.0
```



```
http status code=200
http status code=200
ANALYSING....
5
```

- All the test cases got passed and the expected output is received as the result.

8.2 User Acceptance Testing



User 1

I can see that the application is quite easy to access and use to see the right Universities for myself

I love how there is no registeration or login process, which protects my personal data

The results I got from this software is very close to what I'm eligible for in terms of Universities

User 2

I immediately hoped on to the results after giving my GRE Scores and other related scores

I see that the process is easy and simple and also straight to the point

I got my expected results and I can see that it is quite accurate

9. RESULTS

9.1 Performance Metrics

The data is trained and tested with all three algorithms and out of all KNN gave more accuracy with 90.3 percent and then the KNN with 88.33 percent accuracy. As KNN gave the highest accuracy, all further data predictions are chosen to be followed with KNN.

So, finally a web application is made to give the input parameters of the student and the final prediction is generated and displayed. The background algorithm being used is KNN and the new prediction are keep on adding to the dataset for further more accuracy.

10.ADVANTAGES AND DISADVANTAGES

Advantages: It helps student for making decision for choosing a right college.Here the chance of occurrence of error is less when compared with the existing system.Avoids data redundancy and inconsistency. It is fast, efficient and reliable.

Disadvantages: Machine errors are unavoidable when occurred. (Hardware failure, network failure, others). The predictions made are not 100% accurate but accurate to an acceptable value

11. CONCLUSION

It has been concluded that the software system that we built is successfully executing our aim. The students that are willing to get into a great college, use this website to get more awareness. All this features given to the users at ease, without collecting any of their personal data expect the marks, which we use to train the system, to produce more better and accurate result as we go.

12. FUTURE SCOPE

A powerful web application can be developed where inputs are not given directly instead student parameters are taken by evaluating students through various evaluations and examinations. Technical, analytical, logical, memory based, psychometry and general awareness, interests and skill based tests can be designed and parameters are collected through them so that results will be certainly accurate and the system will be more reliable to use.

Also KNNs have few limitations like overfitting, no pruning, lack of capability to deal with null and missing values and few algorithms have problem with huge number of values. All these can be taken into consideration and even more reliable and more accurate algorithms can be used. Then the project will be more powerful to depend upon and even more efficient to depend upon.

13. APPENDIX

Source Code:

Login.html

```
<!DOCTYPE html>


<html lang="en">

<head>

   <meta charset="UTF-8">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

   <title>Login</title>

   <link rel="stylesheet" href="login-style.css">

   <link href="https://fonts.googleapis.com/css?family=Raleway:700,500,1000&display=swap"
rel="stylesheet">

   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js">
</script>

<script>

   $(function() {

  var people = [];

  $.getJSON('  https://api.thingspeak.com/channels/1013258/feeds.json?results=1', function(data) {

     $.each(data.feeds, function(i, f) {

        var tblRow = "<tr>" + "<td>" + f.created_at + "</td>" +"<td>" + f.entry_id + "</td>" + "<td>"
+ f.field8 + "</td>"+"</tr>"

         $(tblRow).appendTo("#userdata tbody");

    });

  });

});

</script>

</head>

<body>

   <main>
```

```
    <div class="background">
      <div class="text">
        <center style="color:blue">
        <h1>PREDICTING COLLEGES</h1>
</center><br>
<center>
<form  action="https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&"
method="post" target="_blank">


 <div class="form-group">
  <label>GRE Score</label>
  <input type="text" class="form-control" name="field1">
 </div>
 <div class="form-group">
  <label>TOEFL Score</label>
  <input type="text" class="form-control" name="field2">
 </div>
 <div class="form-group">
  <label>UNVERSITY RATING</label>
  <input type="text" class="form-control" name="field3">
 </div>
 <div class="form-group">
  <label>SOP</label>
  <input type="text" class="form-control" name="field4">
 </div>
 <div class="form-group">
  <label>LOR</label>
  <input type="text" class="form-control" name="field5">
 </div>
```

```
  <div class="form-group">
    <label>CGPA</label>
    <input type="text" class="form-control" name="field6">
  </div>
  <div class="form-group">
    <label>RESEARCH</label>
    <input type="text" class="form-control" name="field7">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<table id= "userdata" border="2">
  <thead>
        <th>Date</th>
        <th>S.no</th>
        <th>COLLAGES</th>
         </thead>
     <tbody>
      </tbody>
   </table>
<iframe id='track' frameborder="0" scrolling="no" width="1" height="1">
```

```
</script>

</center>

         </div>

      </div>

   </main>

   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1
" crossorigin="anonymous"></script>

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

</body>

</html>
```

Login-style.css:

```
 {

   margin: 0;

   padding: 0;

   box-sizing: border-box;

   text-decoration: none;

}



body{

   font-family: 'Raleway', sans-serif;

   background: #000;

}



.background{

   background: url(background.jpeg) no-repeat;
 background-position: center top;

   background-size: contain;
height:2200px;
position:relative
   display: flex;
}

.text, .box{
   margin-top:0vh;
   flex: 1;
}
.text{
   margin-left: 0%;
   font-weight: 200px;
color:white;
}
.box{
   margin-right: 25%;
}
.text h1{
   font-size: 70px;
   color: #fff;
   font-weight: 500;
}
.text h2{
   font-size: 70px;
   color: #fff;
   font-weight: 500;
}
 .text p{
   font-size: 20px;
   color: #fff;
   font-weight: 300;
}
 .text p a{
   color: #fff;
   font-weight: 700;
}
 .form{
   background: transparent;
   color: #fff;
   box-sizing: border-box;
   display: flex;
   flex-direction: column;
   width:50%;
}
 input{
   margin: 20px 0;
   padding: 10px;
   background: transparent;
   border: none;
   outline: none;
   color: #fff;
   font-family: 'Raleway', sans-serif;
}
```

```
 .username, .password{
   border-bottom: 1px solid #fff;
}
 .button{
   background: transparent;
   border: 1px solid #fff;
   color: #fff;
   font-size: 18px;
}
 .button:hover{
   background: #000;
   color: #fff;
}
```

College,ipynb

i)

```
import numpy as np
import pandas as pd
from sklearn import metrics
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
data = pd.read_csv('data.csv')
data.head()
data.shape
X = data.iloc[:,:-1]
X.head()
```

```
y = data.iloc[:,-1]
y.head()
print(X)
print(y)
data['Chance of Admit '].value_counts()
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=1)
sns.countplot(x='Chance of Admit ',data=data)
plt.show()
X_train.shape
X_train.head()
y_test.shape
y_test.head()
from sklearn.metrics import accuracy_score
max_accuracy = 0

from sklearn.neighbors import KNeighborsClassifier
for x in range(1,100):
  model = KNeighborsClassifier(n_neighbors=x)
  model.fit(X_train,y_train)
  y_pred = model.predict(X_test)
  current_accuracy = round(accuracy_score(y_pred,y_test)*100,2)
  if(current_accuracy>max_accuracy):
    max_accuracy = current_accuracy
    best_x = x

#print(max_accuracy)
print(best_x)
model = KNeighborsClassifier(n_neighbors=best_x)
```

```
model.fit(X_train,y_train)

y_pred = model.predict(X_test)

filename = 'knn.sav'

pickle.dump(model, open(filename, 'wb'))

acc=(metrics.accuracy_score(y_pred,y_test)*100)

print("Accuracy is:",acc)

cm1 = metrics.confusion_matrix(y_pred,y_test)


total1=sum(sum(cm1))

sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])

print('Sensitivity : ', sensitivity1 )

specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])

print('Specificity : ', specificity1)
```

ii)

```
import pickle

import urllib.request

import json

from time import sleep

while True:

 conn = urllib.request.urlopen("https://api.thingspeak.com/channels/1013258/feeds.json?results=1")

 response = conn.read()

 print ("http status code=%s" % (conn.getcode()))

 data=json.loads(response)

 x=int(data['feeds'][0]['entry_id'])

 y=x

 conn.close()
```

```
while x==y:
  conn = urllib.request.urlopen("https://api.thingspeak.com/channels/1013258/feeds.json?results=1")
  response = conn.read()
  #print ("http status code=%s" % (conn.getcode()))
  data=json.loads(response)
  y=int(data['feeds'][0]['entry_id'])
  conn.close()

conn = urllib.request.urlopen("https://api.thingspeak.com/channels/1013258/feeds.json?results=1")
response = conn.read()
print ("http status code=%s" % (conn.getcode()))
data=json.loads(response)
a=float(data['feeds'][0]['field1'])
b=float(data['feeds'][0]['field2'])
c=float(data['feeds'][0]['field3'])
d=float(data['feeds'][0]['field4'])
e=float(data['feeds'][0]['field5'])
f=float(data['feeds'][0]['field6'])
g=float(data['feeds'][0]['field7'])
conn.close()
filename = 'knn.sav'
loaded_model = pickle.load(open(filename, 'rb'))
person_reports = [[a,b,c,d,e,f,g]]
predicted = loaded_model.predict(person_reports)
print("ANALYSING....")
print(predicted[0])
sleep(15)
```

```
        if predicted[0]==1:

            conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
-VIT

       2-JPR

       3-AGNI")

        elif predicted[0]==2:

            conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
-SREC

       2-KEC

       3-KPR")

        elif predicted[0]==3:

            conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
-KONGU

       2-KCT

       3-HIT")

        elif predicted[0]==4:

            conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
-SASTHRA

       2-SKCET

       3-BIT")

        elif predicted[0]==5:

            conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
-SRM

       2-THIAGARAJAR

       3-NIIT")

        elif predicted[0]==6:

            conn =
urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
 -PSG

            2-CIT

            3-GCT")

              elif predicted[0]==7:

                  conn = 
 urllib.request.urlopen("https://api.thingspeak.com/update?api_key=QNRWO798ZZV2OEIL&field8=1
 -IIT

            2-MIT

            3-ANNA_UNIVERSITY-CHE")
```

app.py

```
from flask import Flask, render_template

app = Flask(__name__)

import os

import subprocess

@app.route(r'/')

def index():

  return render_template('login.html')



if __name__ == '__main__':

  app.run(debug=True)
```

GitHub & Project Demo Link

Github: https://github.com/IBM-EPBL/IBM-Project-52610-1661013317

Project Demo

Link:https://drive.google.com/file/d/1kgFhN74FNkzqaxZedsVGS_FlnTE2-SOn/view?usp=sharing