

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

A PROJECT REPORT

Submitted by

AZHAGARSAMY S [513119104004]

DEVADHARSHINI G [513119104304]

VENGATESH M [513119104046]

JANY DANIEL J [513119104305]

MAYAVAN C [513119104307]

SUBHIKSHA S [513119104312]

TEAM ID : PNT2022TMID29589

INDUSTRY MENTOR : Sandesh P

FACULTY MENTOR : Vanathi S

TABLE OF CONTENTS

1. **INTRODUCTION**
 - 1.1 Project Overview
 - 1.2 Purpose
2. **LITERATURE SURVEY**
 - 2.1 Existing problem
 - 2.2 References

Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
 - 3.1 Empathy Map Canvas
 - 3.2 Ideation & Brainstorming
 - 3.3 Proposed Solution
 - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
 - 4.1 Functional requirement
 - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
 - 5.1 Data Flow Diagrams
 - 5.2 Solution & Technical Architecture
 - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
 - 6.1 Sprint Planning & Estimation
 - 6.2 Sprint Delivery Schedule
 - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
 - 7.1 Feature 1 - UI
 - 7.2 Feature 2 - Flask App
8. **TESTING**
 - 8.1 Test Cases
 - 8.2 User Acceptance Testing
9. **RESULTS**
 - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX** Source Code

GitHub & Project Demo Link

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 short listing 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.

1.2 Purpose

A persons education plays a vital role in their life. While planning for education students often have several questions regarding the courses, universities, job opportunities, expenses involved, etc. Securing admission in their dream university is one of their main concerns. It is seen that often students prefer to pursue their education from universities which have global recognition.

2. LITERATURE SURVEY

When it comes to international students the first choice of the majority of them is the United States of America. With the majority of world highly reputed universities, wide range of courses offered in every sector, highly accredited education system and teaching, scholarships provided to students, best job market and many more advantages make it the dream destination for the international 1 students. According to research, there are above 8 Million international students studying in more than 1700 public and 2500 private universities and colleges across the USA. (Master Portal (2017)).

2.1 Existing problem

Universities take into consideration different factors like score on aptitude based examination like the General Record Examination (GRE), command over the English language is judged based on their score in English competency test like Test Of English as a Foreign Language (TOEFL) OR International English Language Testing System (IELTS), their work experience in same or other fields, the quality of the Letters Of Recommendation (LOR) and the Statement Of Purposedocuments provided by the student etc. Based on the overall profile of the student decision is taken by the universities admission team to admit or reject a particular candidate.

2.2 References

Project Title	Author	Abstract
Recommender system for predicting students admission to a graduate program using machine learning algorithm	Rim marah, Aimad qazdar, Insasaf ei guabassi	. Different application areas adapt and adopt Machine Learning techniques in their systems such as medicine, finance, marketing, business intelligence, health care, etc. In our case, we aim to design a recommender system based on Machine Learning techniques in the field of Education. Thus, the contributions were threefold: The first was to apply several Supervised Machine Learning algorithm
GRADUATE ADMISSION PREDICTION USING MACHINE LEARNING TECHNIQUES	K. Jeevan Ratnakar, G. Koteswara Rao, B. Durga Prasanth Kumar	This paper helps on predicting the eligibility of Indian students getting admission in best university based on their Test attributes like GRE, TOEFL, LOR, CGPA etc. according to their scores the possibilities of chance of admit is calculated
PREDICTING STUDENT UNIVERSITY ADMISSION USING LOGISTIC REGRESSION	Sharan Kumar Paratala Rajagopal	The admission decision depends on criteria within the particular college or degree program.

2.3 Problem Statement Definition

Build an application that predicts the university admission chances of a student powered by machine learning models. Train the model and host it on IBM cloud. The majority of international students studying in the USA are from India and China. In the past decade, India has seen a huge increase in the number of students opting to pursue their education from foreign universities in countries like The USA, Ireland, Australia, Germany, etc. Although there are significant universities and colleges in India, students are finding it difficult to get admission in the highly ranked colleges and also getting a job is a challenge as the ratio of number of students to the number of work opportunities available is quite high. India is one of the leading countries in the number of software engineers produced each year; it becomes tough for the students to find jobs in elite companies due to high competition. This motivates a good number of students to pursue post-graduation in their field. It is seen that the number of students pursuing Masters in Computer Science field from universities in the USA is quite high; the focus of this research will be on these students.

3. IDEATION & PROPOSED SOLUTION

The project aims to develop an application that uses artificial intelligence with the help of a chat bot to customize products for the customers which enhances the fame of an e-commerce store and reduce the time which customers spend on choosing products. The application also uses IBM Cloud storage for storing objects. An application that predicts the university admission chances of a student powered by machine learning models. Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. The primary objective of this research is to develop a system to solve the problems the international students are facing while applying for universities in the USA.

3.1 Empathy Map Canvas

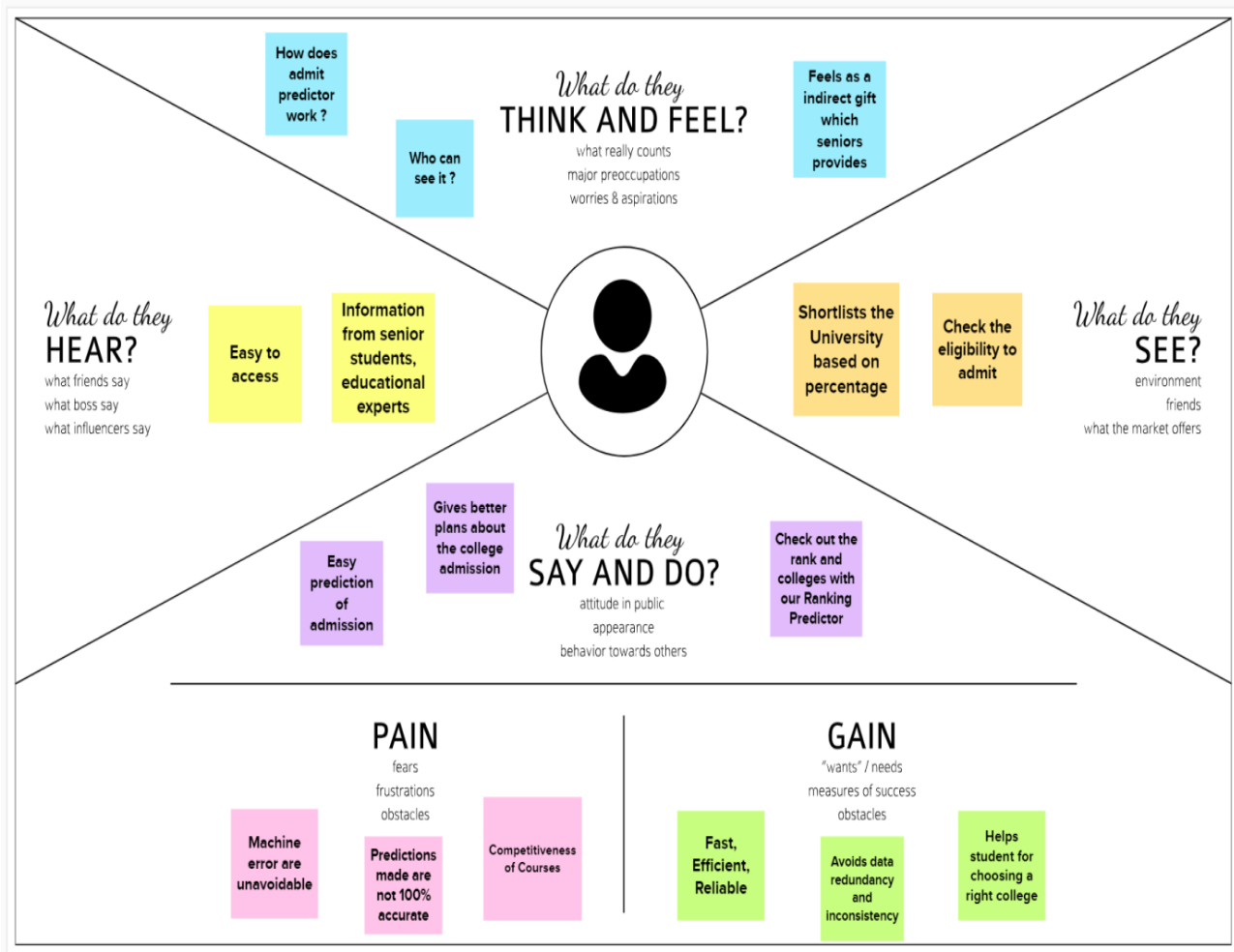
Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1


Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2 Ideation and Brainstorming


Step-1: Team Gathering, Collaboration and Select the Problem Statement


Template





Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

 10 minutes to prepare


 1 hour to collaborate


 2-8 people recommended




Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.


 10 minutes

 **Team gathering**


Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.


 **Set the goal**

Think about the problem you'll be focusing on solving in the brainstorming session.

 **Learn how to use the facilitation tools**


Use the Facilitation Superpowers to run a happy and productive session.


[Open article](#) 




Define your problem statement

What problem are you trying to solve? Define the problem as a How Might We statement to focus of your brainstorm.


 5 minutes


 **PROBLEM**

How might we solve this problem statement?

 **Key rules of brainstorming**

To run an smooth and productive session, follow these key rules of brainstorming:

 **Stay in topic.**



Step-2: Brainstorm, Idea Listing and Grouping

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

AZHAGARSAMY



SUBHIKSHA



MAYAVAN



DEVADHARSHNI



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

⌚ 20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

To ease of
making better
choices of
college before
allotment

Retrive data,from
the database using
functions from
panda's library .The
functions used are
iloc,loc,query,etc.,

It also provides an
analysis based on
the data set used
that shows how the
different parameters
affect chances of
admissions



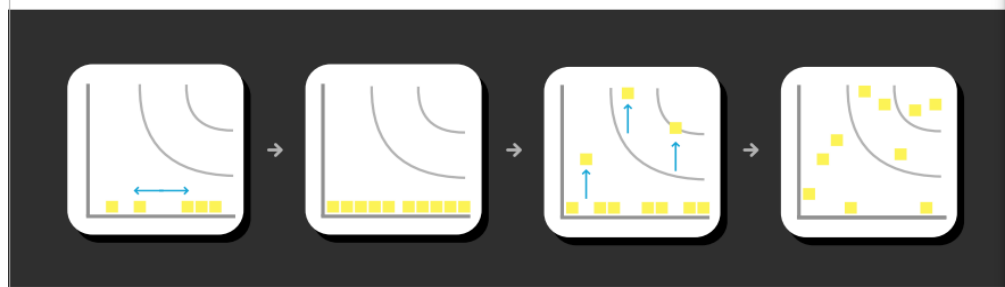
Step-3: Idea Prioritization

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes



3.3 Proposed Solution

SI NO	Parameter	Description
1	Problem Statement (Problem to be solved)	Every year thousands of college graduates apply for the master and PhD programs in US universities from all around the world. Applying to US universities is not an easy task,
2	Idea / Solution description	This issue being a big problem for students have not been solved till now. There are recognized sites which filters the best universities and colleges based on the location, tuition fees, major and degree but none of them have use machine learning algorithm to solve the issue. Hence, we have done this research project to solve that issue to some extent with the use of data mining techniques.
3	Novelty / Uniqueness	University Application process itself being a tedious task Students needs lots of endeavor and determination for completing overall application process. It would definitely be easier for students if they get relief from step of selecting best suited universities and colleges for application.
4	Social Impact / Customer Satisfaction Business Model (Revenue Model)	Results of this project are not applicable to college graduates of each and every major. As there was limitation of information on dataset this system could not predict and recommend universities to students of every major. Nevertheless, the statistical data mining techniques used in this project can be applicable to all majors. If any universities have insufficient data on the major chosen by the student it will return insufficient data for prediction to the user
5	Business Model (Revenue Model)	From this project, financially can earn from the students admission fees but while they want to first select in their selected college in prediction. Although which is done by this project for prediction. In this project, this problem has been addressed by modeling a recommender system based on various classification algorithms. The required data was obtained from thegradcafe.com. Based on this data set, various models were trained and one best and some other similar properties carrying universities are suggested for the students such that it maximizes the chances of a student getting an admit from that university list
6	Scalability of the Solution	In this project, this problem has been addressed by modelling a recommender system based on various classification algorithms. To predict the best University for the particular student his/her GPA score, GRE (Verbal and Quant) Score, TOEFL score has been used as attributes for classification.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Profile	Complete user profile by providing the Student Academic details.
FR-2	User Search	Search for desired University based on their Academic Performance and eligibility criteria.
FR-3	User Preference	Search for Universities based on their location preference.
FR-4	Result	The list of universities is filtered based on the eligibility of the students where the order of the list will be based on the ratings of the university.

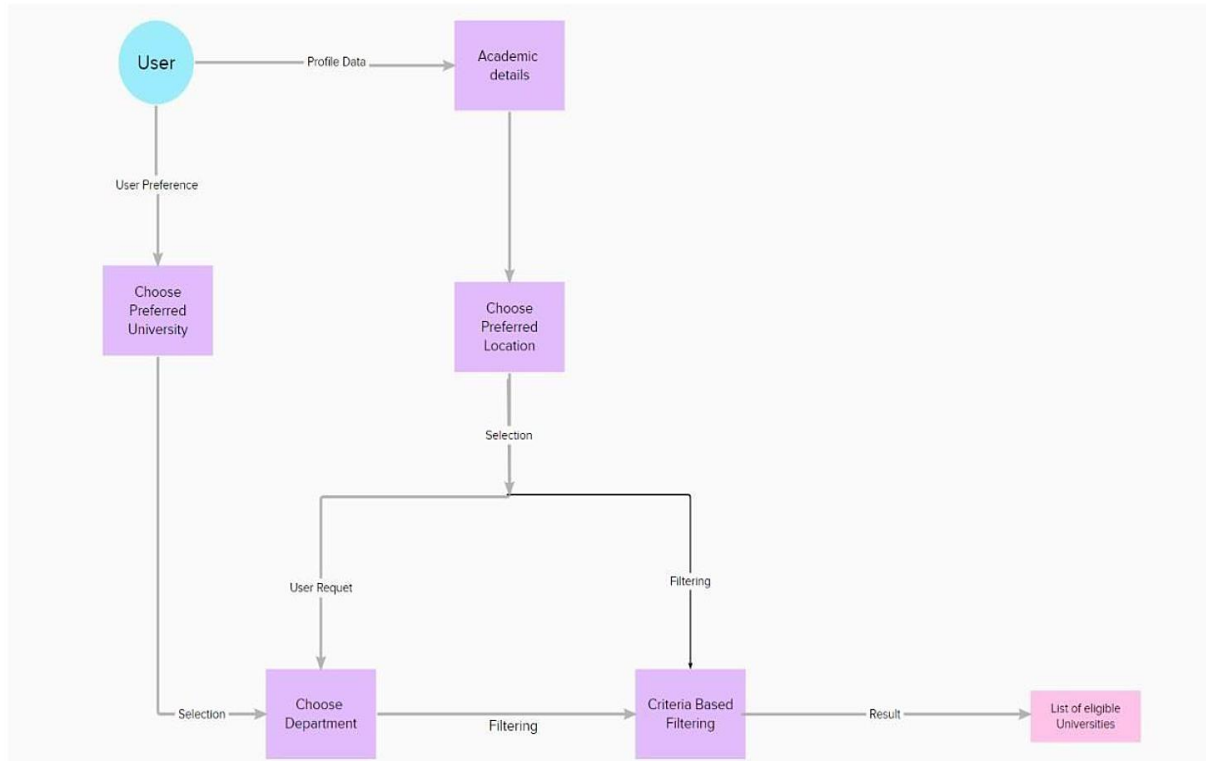
4.2 Non-Functional requirements

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Filters the universities based on the user profile.
NFR-2	Security	User details are secured from un authorised parties.
NFR-3	Reliability	The users can find universities based on their preferred location and results.
NFR-4	Performance	The website will provide the list of universities within 30 seconds.
NFR-5	Availability	Students across India can access the website anytime.
NFR-6	Scalability	The solution will be helpful for the students in India to know the details about universities they are eligible.

5. PROJECT DESIGN

5.1 Data Flow Diagram

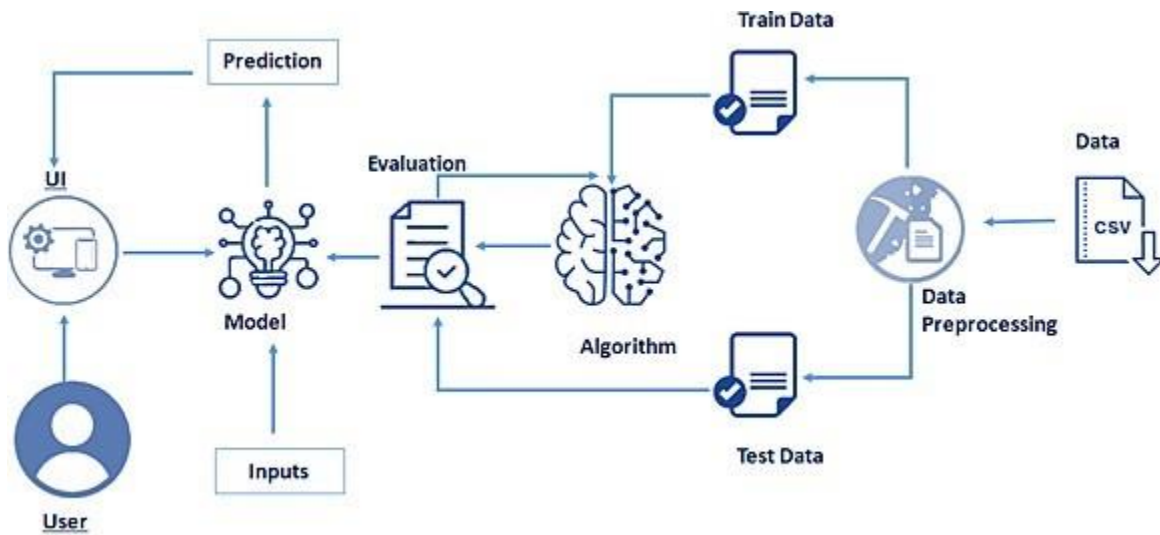


5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

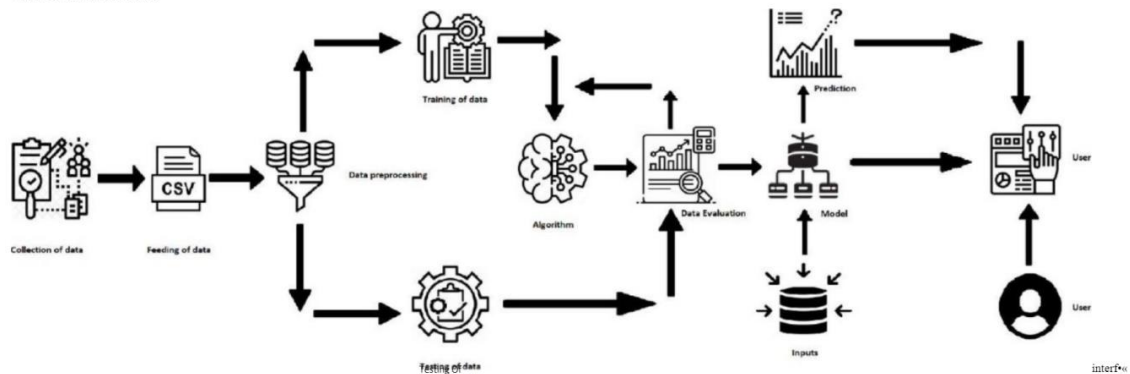
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements. Provide specifications according to which the solution is defined, managed, and delivered

Solution Architecture:



Technical Architecture:

PNT2022TMID29589



5.3 User Stories

A user story is an informal, natural language description of features of a software system. They are written from the perspective of an end user or user of a system, and may be recorded on index cards, Post-it notes, or digitally in project management software. [1] Depending on the project, user stories may be written by different stakeholders like client, user, manager, or development team.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Profile	USN-1	As a user, I can Give my academic information in the profile section	I can access my dashboard	High	Sprint-1
		USN-2	As a user, I will be able to select a location that I prefer	I can receive the list of location in the dropdown to select	High	Sprint-1
	Search	USN-3	As a user I can search for my preferred university	I can use the search bar	Medium	Sprint-2
	User Preference	USN-4	As a user, I can select my preferred university from the list to check my eligibility for the particular university	I can use the dropdown list provided to select the university	Medium	Sprint-2
		USN-5	As a user, I can select my preferred location	I can select my preferred location	High	Sprint-1
		USN-6	As a user, I will be able to select my preferred department	I can select a department from the dropdown list	Medium	Sprint-1
	Result	USN-7	As a user, I can view the list of universities that I am eligible in accordance to my preferred location	I can view the list of universities filtered by the model	High	Sprint-3
		USN-8	As a user, I can access the link to the university that I am eligible from the list	I can access the university link	Medium	Sprint-3
		USN-9	As a user, I can access the location link of the university that I am eligible from the list	I can access the university location link	Low	Sprint-3
		USN-10	From the list of universities, I can select and view the eligibility for the particular university	I can view the eligible university	Medium	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & 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 will be able to register my application by entering my email, password, and confirming my password.	2	High	Azhagarsamy
Sprint-1		USN-2	As a user, I will be able to receive an email confirmation after registration.	1	High	Jany daniel
Sprint-2		USN-3	As a user, I can register for the application through Gmail.	2	Low	Mayavan , Vengatesh
Sprint-1		USN-4	As a user, I can register for the application by entering details by self.	2	Medium	Subhiksha
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Deva dharshini
	Dashboard					

6.2 Sprint Delivery Schedule

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

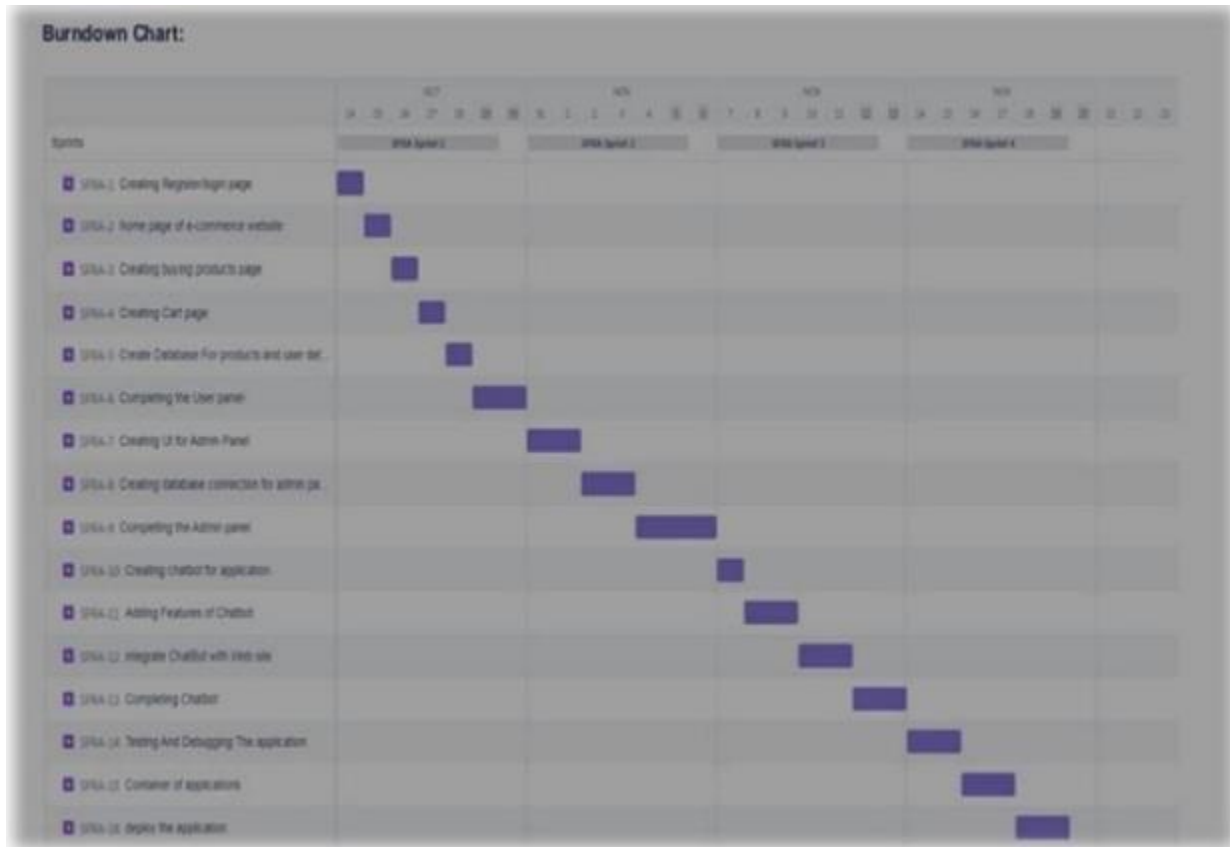
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

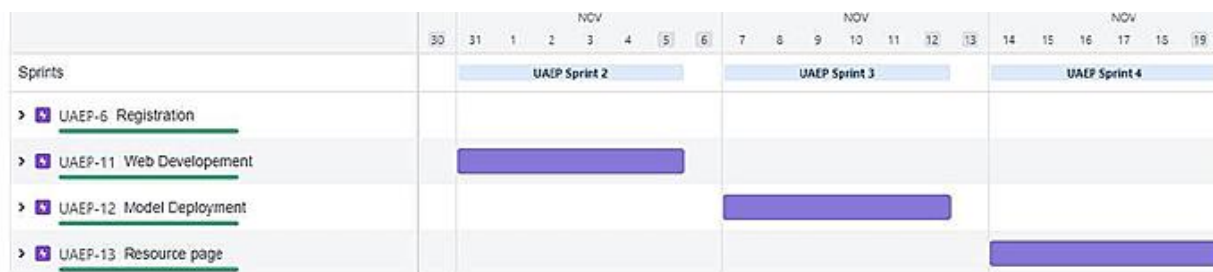
sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

6.3 Reports from JIRA



BURNDOWN CHART

Road map:



7. CODING & SOLUTIONING

7.1 Feature 1 -UI

The following is the UI code for the application
index.html:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <SCRIPT language=Javascript>
5  <!--
6  function check(e, value) {
7    //Check Charater
8    var unicode = e.charCode ? e.charCode : e.keyCode;
9    if (value.indexOf(".") != -1)
10     if (unicode == 46) return false;
11    if (unicode != 8)
12     if ((unicode < 48 || unicode > 57) && unicode != 46) return false;
13  }
14  //-->
15 </SCRIPT>
16 <meta name="viewport" content="width=device-width, initial-scale=1">
17 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
  awesome/4.7.0/css/font-awesome.min.css">
18 <style>
19 body {
20   font-family: Arial, Helvetica, sans-serif;
21 }
22
23 * {
24   box-sizing: border-box;
25 }
26
27 /* style the container */
28 .container {
29   position: relative;
30   border-radius: 5px;
31   background-color: #f2f2f2;
32   padding: 20px 0 30px 0;
33 }
34
```

```

35 /* style inputs and link buttons */
36 input,
37 .btn {
38     width: 100%;
39     padding: 12px;
40     border: none;
41     border-radius: 4px;
42     margin: 5px 0;
43     opacity: 0.85;
44     display: inline-block;
45     font-size: 17px;
46     line-height: 20px;
47     text-decoration: none; /* remove underline from anchors */
48 }
49
50 input:hover,
51 .btn:hover {
52     opacity: 1;
53 }
54
55
56 /* style the submit button */
57 input[type=submit] {
58     background-color: #4c66af;
59     color: white;
60     cursor: pointer;
61 }
62
63 input[type=submit]:hover {
64     background-color: #45a049;
65 }
66
67 /* Two-column layout */
68 .col {
69     float: left;
70     width: 50%;
71     margin: auto;
72     padding: 0 50px;
73     margin-top: 6px;
74 }
75
76 /* Clear floats after the columns */
77 .row:after {
78     content: "";
79     display: table;
80     clear: both;

```

```

81 }
82
83 /* vertical line */
84 .vl {
85     position: absolute;
86     left: 50%;
87     transform: translate(-50%);
88     border: 2px solid #ddd;
89     height: 490px;
90 }
91
92 /* text inside the vertical line */
93 .vl-innertext {
94     position: absolute;
95     top: 50%;
96     transform: translate(-50%, -50%);
97     background-color: #f1f1f1;
98     border: 1px solid #ccc;
99     border-radius: 50%;
100    padding: 8px 10px;
101 }
102
103 /* hide some text on medium and large screens */
104 .hide-md-lg {
105     display: none;
106 }
107
108 /* bottom container */
109 .bottom-container {
110     text-align: center;
111     background-color: #666;
112     border-radius: 0px 0px 4px 4px;
113 }
114
115 /* Responsive layout - when the screen is less than 650px wide, make the two columns
    stack on top of each other instead of next to each other */
116 @media screen and (max-width: 650px) {
117     .col {
118         width: 100%;
119         margin-top: 0;
120     }
121     /* hide the vertical line */
122     .vl {
123         display: none;
124     }
125     /* show the hidden text on small screens */

```

```

126 .hide-md-lg {
127     display: block;
128     text-align: center;
129 }
130}
131</style>
132</head>
133<body>
134
135<div class="container">
136     <form action="{ {url_for('predict')}} " method="post">
137         <div class="row">
138             <marquee class="bottom-container"><h2>Graduate Admission
139                 Prediction</h2></marquee>
140             <div class="v1">
141                 <span class="v1-innertext"></span>
142             </div>
143             <div class="col">
144                 <p><b>In this project, I build a linear regression model to predict the chance of
145                     admission into a particular university based on student's profile.</p></b>
146                 <h3>Instructions for Input Features</h3>
147                 <ul>
148                     <li>GRE Score (out of 340)</li>
149                     <li>TOEFL Score (out of 120)</li>
150                     <li>University Rating (out of 5)</li>
151                     <li>Statment of Purpose { SOP } (out of 5)</li>
152                     <li>Letter of Recommendation { LOP } Strength (out of 5)</li>
153                     <li>Undergraduate CGPA (out of 10)</li>
154                     <li>Research Experience (Either 0 or 1)</li>
155                 </ul>
156             </div>
157             <div class="col">
158                 <div class="hide-md-lg">
159                     <input type="number" name="GRE Score" placeholder="GRE Score"
160                         required="required" min="0" max="340"/>
161                     <input type="number" name="TOEFL Score" placeholder="TOEFL
162                         Score" required="required" min="0" max="120"/>
163                     <input type="number" name="University Rating"
164                         placeholder="University Rating" required="required" min="1" max="5"/>

```

```

165         <input type="number" name="SOP" placeholder="SOP"
        required="required" onkeypress="return check(event,value)" step="0.1" min="1"
        max="5"/>
166         <input type="number" name="LOR" placeholder="LOR"
        required="required" onkeypress="return check(event,value)" step="0.1" min="1"
        max="5"/>
167         <input type="number" name="CGPA" placeholder="CGPA"
        required="required" onkeypress="return check(event,value)" step="0.01" min="1"
        max="10"/>
168         <input type="number" name="Research" placeholder="Research"
        required="required" min="0" max="1"/>
169
170         <input type="submit" value="Predict"></input>
171
172         <h4 style="text-align: center;">{{ prediction_text }}</h4>
173     </div>
174 </div>
175 </form>
176</div>
177
178<div class="bottom-container">
179 <div class="row">
180   <marquee><h3>prediction Of Admit</h3></marquee>
181 </div>
182</div>
183
184</body>
185</html>

```


output:

The screenshot shows a web browser window with a single tab titled 'New Tab' at the address '127.0.0.1:5000/'. The web application has a dark header with the title 'Graduate Admission'. Below the header, on the left, is a text block stating: 'In this project, I build a linear regression model & randomforestclassifier model to predict the chance of admission into a particular university based on student's profile.' This is followed by 'Instructions for Input Features' which lists: GRE Score (out of 340), TOEFL Score (out of 120), University Rating (out of 5), Statement of Purpose (SOP) (out of 5), Letter of Recommendation (LOR) Strength (out of 5), Undergraduate CGPA (out of 10), and Research Experience (Either 0 or 1). To the right of this text is a vertical slider. Further right is a form with input fields for 'GRE Score', 'TOEFL Score' (with a 'Please enter a number.' error message), 'University Rating', 'SOP', 'LOR', 'CGPA', and 'Research'. Each field has a small circular icon to its right. At the bottom of the form is a blue 'Predict' button. Below the form is a dark bar with the text 'Prediction of Admit'. The Windows taskbar at the bottom shows the date as 11/20/2022 and time as 7:38 PM.

7.2 Feature 2 - Flask App

```
1 import pandas as pd
2 from flask import Flask, request, jsonify, render_template
3 import pickle
4 import requests
5 from math import ceil
6
7 API_KEY = "QfeBSqnNqjSmGFdJ_DC2d6rWSA21J21p0dDt9kDCERhm"
8 token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
9 data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
10 mltoken = token_response.json()["access_token"]
11 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
12 app = Flask(__name__)
13
14
15 @app.route('/')
16 def home():
17     return render_template('index.html')
18
19 @app.route('/predict', methods=['GET', 'post'])
20 def predict():
```

```

21
22 GRE_Score = int(request.form['GRE Score'])
23 TOEFL_Score = int(request.form['TOEFL Score'])
24 University_Rating = int(request.form['University Rating'])
25 SOP = float(request.form['SOP'])
26 LOR = float(request.form['LOR'])
27 CGPA = float(request.form['CGPA'])
28 Research = int(request.form['Research'])
29
30 final_features = [GRE_Score, TOEFL_Score, University_Rating, SOP, LOR, CGPA,
Research]
31 x = ['GRE_Score', 'TOEFL_Score', 'University_Rating', 'SOP', 'LOR', 'CGPA',
'Research']
32 payload_scoring = { "input_data": [{"fields":x, "values": [final_features]]}
33 response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/72590142-1e6c-4a84-ab58-
8fe106dadb4e/predictions?version=2022-11-19',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
34 predictions=response_scoring.json()['predictions'][0]['values'][0][0]
35 predict=ceil(predictions[0])
36
37 if predict>50:
38     predict=ceil(predictions[0]*1.44927536)
39 else:
40     predict=ceil(predictions[0])
41 if predict>50:
42     return render_template('index.html', prediction_text='You have a {}%Chance
to Get Admission '.format(predict))
43 else:
44     return render_template('index.html', prediction_text='You have NO chance of {}%
to Get Admission '.format(predict))
45
46 if __name__ == "__main__":
47     app.run(debug=True)
48

```

output:

New Tab x 127.0.0.1:5000/ +

127.0.0.1:5000/predict

Graduate Admission Pre

In this project, I build a linear regression model & randomforestclassifier model to predict the chance of admission into a particular university based on student's profile.

Instructions for Input Features

- GRE Score (out of 340)
- TOEFL Score (out of 120)
- University Rating (out of 5)
- Statment of Purpose (SOP) (out of 5)
- Letter of Recommendation (LOR) Strength (out of 5)
- Undergraduate CGPA (out of 10)
- Research Experience (Either 0 or 1)

GRE Score

TOEFL Score

University Rating

SOP

LOR

CGPA

Research

Predict

You have a 60%Chance to Get Admission

Prediction of Admit

77°F Cloudy Search 7:40 PM 11/20/2022

8.TESTING

8.1 Test Cases

Case 1- For eligible students

Graduate Admission Pre

In this project, I build a linear regression model & randomforestclassifier model to predict the chance of admission into a particular university based on student's profile.

Instructions for Input Features

- GRE Score (out of 340)
- TOEFL Score (out of 120)
- University Rating (out of 5)
- Statment of Purpose (SOP) (out of 5)
- Letter of Recommendation (LOR) Strength (out of 5)
- Undergraduate CGPA (out of 10)
- Research Experience (Either 0 or 1)

GRE Score

TOEFL Score

University Rating

SOP

LOR

CGPA

Research

Predict

You have a 60%Chance to Get Admission

Prediction of Admit

77°F Cloudy

Search

ENG IN

7:40 PM 11/20/2022

Case 2- For Not eligible students

Graduate Admission Pr

In this project, I build a linear regression model & randomforestclassifier model to predict the chance of admission into a particular university based on student's profile.

Instructions for Input Features

- GRE Score (out of 340)
- TOEFL Score (out of 120)
- University Rating (out of 5)
- Statment of Purpose (SOP) (out of 5)
- Letter of Recommendation (LOR) Strength (out of 5)
- Undergraduate CGPA (out of 10)
- Research Experience (Either 0 or 1)

GRE Score

TOEFL Score

University Rating

SOP

LOR

CGPA

Research

Predict

You have NO chance of 48% to Get Admission

Prediction of Admit

8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done. The User Acceptance of this product is not surveyed enough to give a solid conclusion. The theoretical and hypothetical acceptance is calculated to be high enough to conclude that this product is usable and valuable.

Defect Analysis:

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	1	0	0	3
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	2	1	0	0	3
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	4	2	0	0	6

Test Case Analysis:

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	13	0	0	13
Client Application	11	0	0	11
Security	2	0	0	2
Outsource Shipping	0	0	0	0
Exception Reporting	2	0	0	2
Final Report Output	11	0	0	11
Version Control	1	0	0	1

9. RESULTS

9.1 PERFORMANCE METRICS

Performance Testing - Machine Learning:

S.No.	Parameter	Values	Output
1.	Metrics	Regression Model: MAE - , MSE - , RMSE - , R2 score -	Mean Absolute Error : 0.039428 Mean Square Error : 0.002954 Root Mean Square Error : 0.054 R2 Score :0.85685
2.	Comparing R2 scores for different regression models	Multiple Linear Regression, Random Forest Regression	Multiple Linear Regression's Score = 0.85685 Random Forest Regression's Score = 0.8229

10. ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- Avoids data redundancy and inconsistency.
- It is fast, efficient and reliable.
- 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

10.2 DISADVANTAGES

- Machine errors are unavoidable when occurred. (Hardware failure, network failure, others).
- Reach to geographically scattered student.
- Reducing time in activities
- Paperless admission with reduced man power
- Operational efficiency
- The predictions made are not 100% accurate but accurate to an acceptable value.

11. CONCLUSION

A model was developed to determine the admission of a student to the interested universities. The following parameters were taken into consideration: GRE Score, TOEFL Score, University Ranking, SOP, LOR, CGPA. From the validations, we can find out that the above parameters greatly contributed in determining the “Chance of Admit” into an university. Different models - Multiple Linear Regression, Random Forest Regression were taken into consideration. Out of the 2 models, Multiple Linear Regression output formed other models with a R2 score of 0.808. Hence Multiple Linear Regression was adopted in predicting the results. The project uses a Linear regressor to predict the output and a web application is built to make the UI more accessible and easy using various technologies such as python, React JS, HTML5, CSS, Flask, Scikit, Mat plot, Numpy, Pandas, Seaborn and other libraries. After the deployment of the web application, it can be accessed from anywhere with internet connection. This project reduces the long hours of analysis to predict the eligibility of the admission to a rated university.

12. FUTURE SCOPE

In future we would like to enhance the existing model in such a way that consumer feels the same way when purchasing in store using Virtual reality and other upcoming technologies. Research to improve the accuracy of the system is under progress. Develop a community consisting of faculty, alumni and aspirants to get to know about the university more .

Get in touch with grad-schools' and professors and determine other important factors that play a key role in increasing the chances of admission.

13. APPENDIX

GitHub Link : [click here to link](#)

Project Demo Link : [demo video](#)