

# **UNIVERSITY ADMIT ELIGIBILITY PREDICTOR**

**IBM PNT2022TMID02243**

## **TEAM MEMBERS:**

Roshini V

Sangamithra V

Shalini M

Sharath Srinivas D

## **Project Report**

### **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

### **2. LITERATURE SURVEY**

- 2.1 References
- 2.2 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
- 7.2 Feature 2
- Database Schema (if Applicable)

### **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**

- 13.1 Source Code
- 13.2 GitHub & Project Demo Link

## **1. INTRODUCTION**

### **1.1 Project Overview**

Students frequently have concerns over their opportunities to enrol in university programmes. This project's goal is to assist students in narrowing down their choices of institutions by improving their profiles. The outcome that was projected provides them with a reasonable estimate of their prospects of being accepted into a certain university. Students who are presently preparing or who will be prepared could also benefit from this analysis because it should assist them to obtain a better picture.

### **1.2 Purpose**

Students will be able to check if they are eligible to enrol in university programmes or not.

## 2. LITERATURE SURVEY

### 2.1 References

| S. No | Paper  | Year | Citation   | Methodologies used  |
|-------|--|------|--|---|
| 1     | Graduate Admission Chance Prediction Using Deep Neural Network   | 2020 | 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)  | The DNN model has been compared with existing methods in terms of different performance metrics including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), R-squared score. It has shown the most promising result that includes an R-squared score of 0.8538 and MSE of 0.0031.              |
| 2     | A Statistical Approach to Graduate Admissions' Chance Prediction | 2020 | Navoneel Chakrabarty, Siddhartha Chowdhury, Srinibas Rana, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 , Volume: 09 Issue: March 2020 | The developed models are gradient boosting regressor and linear regression model. Gradient boosting regressor have to score of 0.84. That surpassing the performance of linear regression model. They computed different other performance error metrics like mean absolute error, mean square error, and root mean square error. |

|   |   |      |  |  |
|---|---|------|--|--|
| 3 | Literature Survey on Student's Performance Prediction in Education using Data Mining Techniques | 2017 | Mukesh Kumar, Disha Handa, International Journal of Education and Management Engineering 6(6):40-49<br>DOI:10.5815/ijme.2017.06.05   | Article aims to provide a great knowledge and understanding of different data mining techniques which have been used to predict the student progress and performance and hence how these prediction techniques help to find the most important student attribute for prediction. It wants to improve the performance of the student in academic by using best data mining techniques |
| 4 | Prediction for University Admission using Machine Learning                                      | 2020 | Chithra Apoorva D A, Malepati ChanduNath, Peta Rohith, Bindu Shree.S, Swaroop.S, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020 | Problem Understanding, Data Understanding, Data Preparation, Building models, Evaluation   |

|   |   |      |   |  |
|---|---|------|---|--|
| 5 | Machine Learning Approach for Predicting Student Enrollment in the University | 2021 | 2021 XXX International Scientific Conference Electronics (ET), Date of Conference: 15-17 September 2021Electronic ISBN:978-1-6654-4518-4, Print on Demand(PoD) ISBN:978-1-6654-4519-1 | The RepTree algorithm performed the best with a True Positive (TP) rate of 0.902 and a False Positive (FP) rate of 0.148. The algorithm with the lowest performance was NaiveMulti with a TP rate of only 0.790. However, the range between the worst and the best-performing algorithms was 14.18%. |
|---|---|------|---|--|

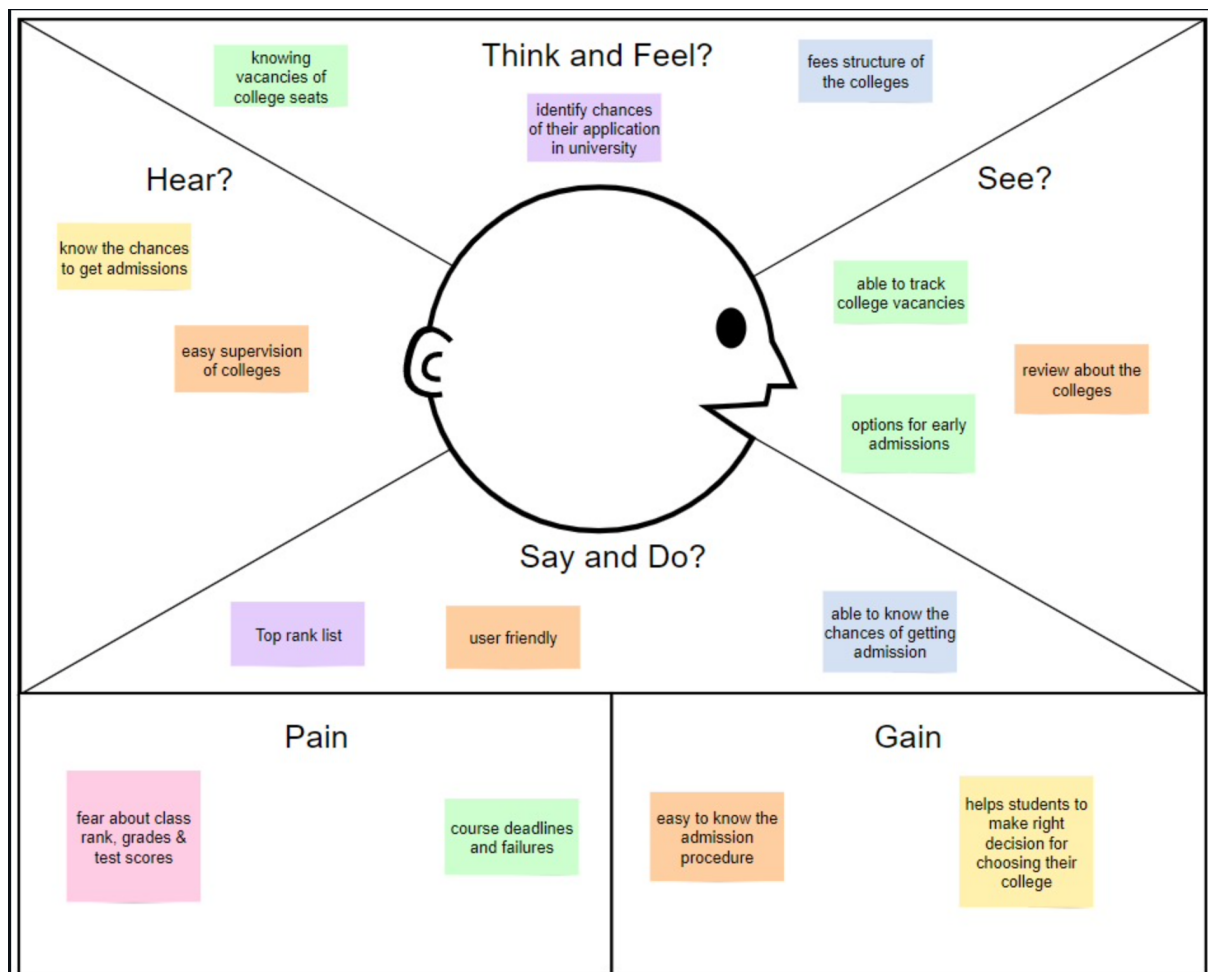
## 2.2 Problem Statement Definition

Students frequently have concerns over their opportunities to enrol in university programmes. This project's goal is to assist students in narrowing down their choices of institutions by improving their profiles. The outcome that was projected provides them with a reasonable estimate of their prospects of being accepted into a certain university. Students who are presently preparing or who will be prepared could also benefit from this analysis because it should assist them to obtain a better picture.

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



## 3.2 Ideation & Brainstorming

1

### Define your problem statement

University Admit Eligibility Predictor

U

PROBLEM

How we might predict the students admission in universities?

2

### Brainstorm

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

🕒 10 minutes

#### TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Shalini

1. Academic Record
2. Recommendations
3. Entrance Mark

Roshini

1. Cut Off
2. Seat Availability
3. Rankings

Sengemithra

1. Fee Expectations
2. Distance
3. Placement

Sharath

1. Student Finances
2. Student preference
3. Extra Curricular



3

### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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

#### Based on Fees



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

#### Based on Academics



#### Based on preference



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





### After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

---

#### Quick add-ons



##### Share the mural

Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.



##### Export the mural

Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

---

#### Keep moving forward



##### Strategy blueprint

Define the components of a new idea or strategy.

[Open the template →](#)



##### Customer experience journey map

Understand customer needs, motivations, and obstacles for an experience.

[Open the template →](#)



##### Strengths, weaknesses, opportunities & threats

Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

[Open the template →](#)



[Share template feedback](#)

### 3.3 Proposed Solution

| S.No. | Parameter                                | Description  |
|-------|--|--|
| 1     | Problem Statement (Problem to be solved) | Many students worry and wonder if they will get into the university they desire. In rare cases, students are not able to attend the universities of their choice. They won't have time to submit applications to other colleges at the last minute. The student feels a lot of stress as a result of this. It would be preferable if students could find out in advance if they can get into the university of their choice so that they can be ready for the worst-case scenario. |
| 2.    | Idea / Solution description              | A program based on machine learning algorithm to classify the result with the help of the given details like academic qualifications, GRE scores, TOEFL scores etc.  |
| 3.    | Novelty / Uniqueness                     | High accurate machine learning algorithm without compromising the features and enhanced UI   |
| .4.   | Social Impact / Customer Satisfaction    | Really helpful and more accurate, as judged by customer feedback and prediction accuracy of more than 90%  |
| 5.    | Business Model (Revenue Model)           | The goals of the research were initially determined in the process of spending a significant amount of time comprehending the problem statement by analysing the concerns of students over the current application procedure thereby turning this into a profitable business model concerning students.  |
| 6.    | Scalability of the Solution              | The application is extremely user-friendly and accessible from anywhere in the globe without the need to register or deal with any additional hassles. It is open-sourced, and there are no restrictions for specific users.   |

### 3.4 Problem Solution fit

## Problem-Solution fit canvas 2.0

Purpose / Vision

|                         |   |  |   |  |
|-------------------------|---|--|---|--|
| Define CS, fit into CC  | <b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span><br>Who is your customer?<br>I.e. working parents of 0-5 y.o. kids<br><br>Students- The most favourable and beneficial segment of the users are the students. Students, in particular, who want to pursue higher degrees in foreign colleges, notably those in the United States.<br><br>Administrators - Aids the university's administration officials in sorting and eliminating applications | <b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span><br>What constraints prevent your customers from taking action or limit their choices of solutions? I.e. spending power, budget, no cash, network connection, available devices.<br><br>• Fluctuating trends and parameters<br>• Ambiguity in the legitimacy of data   | <b>5. AVAILABLE SOLUTIONS</b> <span>AS</span><br>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<br><br>The project Engineering Admission Predictor System an application where students can register with their details for prediction and the administrator can allot the seats manually.<br><br>Pros:<br>• Avoids data redundancy<br>• Very user-friendly.<br>• Easy accessibility of data<br><br>Cons:<br>• Requires active internet connection<br>• inaccurate results                               | Explore AS, differentiate  |
|                         | Focus on J&P, tap into BE, understand RC  | <b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span><br>Which jobs-to-be-done (or problems) do you address for your customers?<br>There could be more than one; explore different sides.<br><br>• Provide a free examination of how different academic characteristics affect university acceptance;<br>• Grant administrator access to all supplied records<br><br>Other aspects of the system that are desired:<br>• Each profile's privacy and security must be upheld<br>• Every account is password-protected.   | <b>9. PROBLEM ROOT CAUSE</b> <span>RC</span><br>What is the real reason that this problem exists?<br>What is the back story behind the need to do this job?<br>I.e. customers have to do it because of the change in regulations.<br><br>• No clear & certain cutoffs<br>• Distinct acceptance parameters adopted by diverse universities<br>• No legitimate or reliable source to vouch for the acceptance   |  |
| Identify strong TR & EM |   | <b>3. TRIGGERS</b> <span>TR</span><br>What triggers customers to act? I.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.<br><br>The implementation of our model would be necessitated by the ongoing hype for earning a degree overseas and uncertainty surrounding it<br><br><b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span><br>How do customers feel when they face a problem or a job and afterwards?<br>I.e. lost, insecure > confident, in control - use it in your communication strategy & design.<br><br>Doubtful, Fear > Confidence, Calmness | <b>10. YOUR SOLUTION</b> <span>SL</span><br>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.<br>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.<br><br>The goal of this project is to help profile shortlisted college students. Predicted results give them a good idea of their likelihood of getting into a particular college. This analysis is also useful for students preparing or planning to prepare for a better image must. | <b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span><br><b>8.1 ONLINE</b><br>What kind of actions do customers take online? Extract online channels from #7<br><br>Prediction results and detailed analysis of the same<br><br><b>8.2 OFFLINE</b><br>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.<br><br>• Download the report and suggestions<br>• Data could be made shareable |

## 4. REQUIREMENT ANALYSIS

#### 4.1 Functional requirement

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task)   |
|--------|-------------------------------|--|
| FR-1   | User Details                  | Submit the Documents: <ul style="list-style-type: none"><li>• GRE or/and TOEFL score sheet</li><li>• Curriculum Vitae (CV)</li><li>• Statement of Purpose (SoP)</li><li>• Letter of Recommendation (LOR)</li></ul>   |
| FR-2   | User Requirements             | <ul style="list-style-type: none"><li>• Relevant documents and details are uploaded in the appropriate location on the website.</li><li>• Based on the uploads, the system would collect all the necessary information.</li><li>• The candidate's list of all possible universities would be displayed based on the collected information.</li></ul> |

#### 4.2 Non-Functional requirements

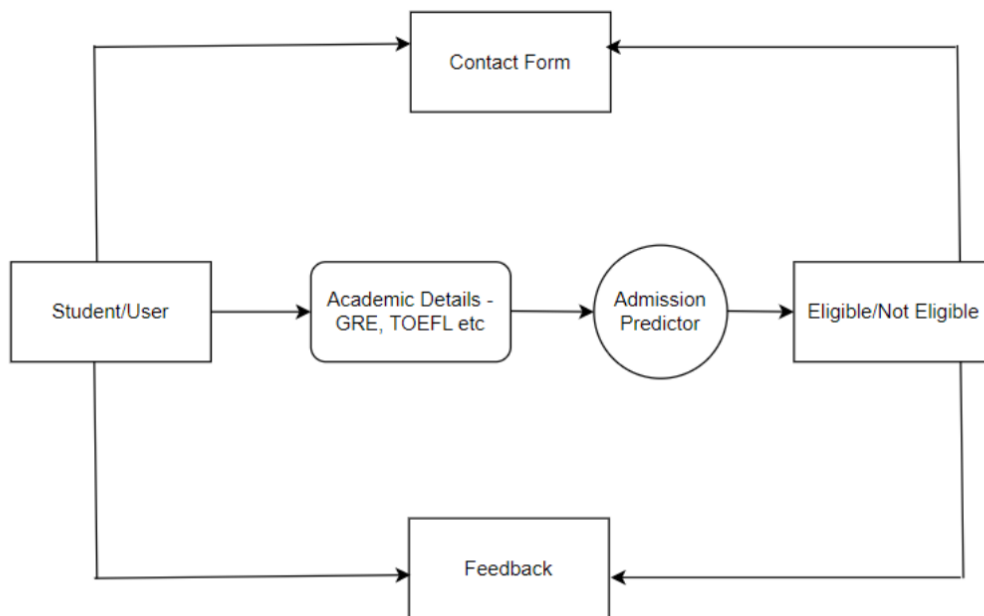
| FR No. | Non-Functional Requirement | Description   |
|--------|----------------------------|---|
| NFR-1  | Usability                  | <ul style="list-style-type: none"><li>• Easy to use and self-explaining website.</li><li>• Easy navigation between pages.</li><li>• Simple structure for faster access.</li></ul> |
| NFR-2  | Reliability                | <ul style="list-style-type: none"><li>• Ensembling the outputs of various ML models.</li></ul>  |
| NFR-3  | Performance                | <ul style="list-style-type: none"><li>• Web Based Application.</li><li>• Ability to indicate user inputs of erroneous data types.</li></ul>                                       |

|       |                     |  |
|-------|---------------------|--|
| NFR-4 | <b>Availability</b> | <ul style="list-style-type: none"> <li>• Application is available 24 / 7 as it is hosted on IBM cloud.</li> <li>• Simple web browser is enough to access the website.</li> </ul> |
| NFR-5 | <b>Scalability</b>  | <ul style="list-style-type: none"> <li>• Can be extended for other types of loans.</li> <li>• Aadhar and PAN verification can also be implemented.</li> </ul>                    |

## 5. PROJECT DESIGN

### 5.1 Data 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.



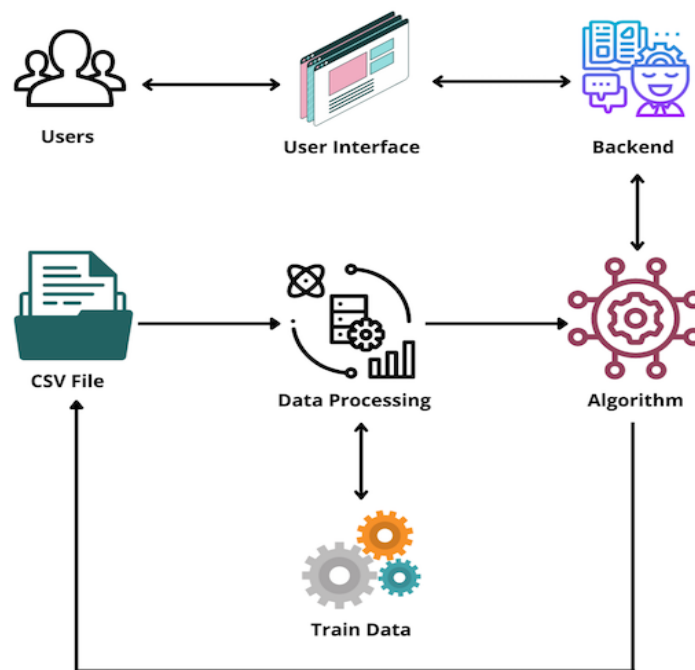
## 5.2 Solution & Technical Architecture

### Solution Architecture:

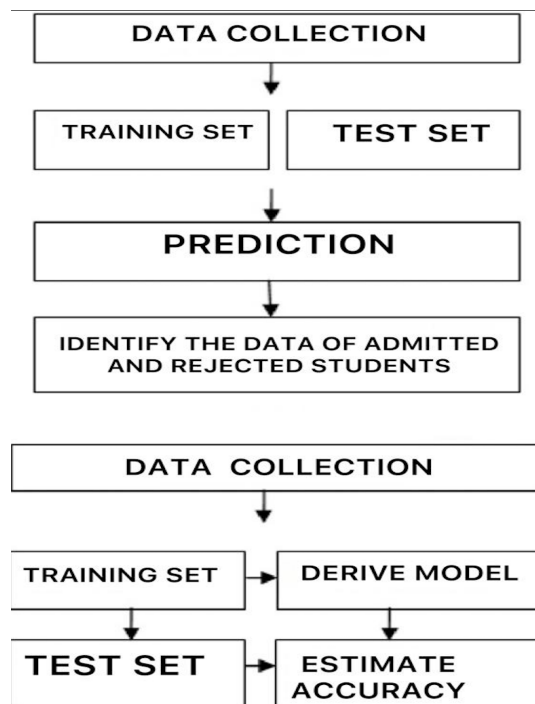
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- 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.





### Technical Architecture:



### 5.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task                                       | Acceptance criteria                              | Priority | Release  |
|-----------|-------------------------------|-------------------|---|--|----------|----------|
| Student   | Dashboard                     | USN-1             | User can view and enter details in the website.         | I can access the university landing page         | Medium   | Sprint 1 |
|           |                               | USN-2             | Queries can be filled, and feedback can be given.       | I can fill and submit the contact form           | Low      | Sprint 2 |
|           | Admission                     | USN-3             | Admission eligibility can be predicted for universities | I can get result as either eligible/not eligible | High     | Sprint 1 |
|           | Updates                       | USN-4             | Updates can be given                                    | I can view the updates                           | Low      | Sprint 3 |
|           | Feedback                      | USN-5             | As a user, I can give a positive or negative feedback   | can give either good or bad feedback             | Low      | Sprint 4 |

## 6. PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation

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 | 10           | High     | Roshini V<br>Sangamithra V<br>Shalini M<br><br>Sharath Srinivas D |
| Sprint-2 | Login                         | USN-3             | Visualise the dataset  | 5            | Low      | Roshini V<br>Sangamithra V<br>Shalini M<br><br>Sharath Srinivas D |
| Sprint-2 | Shortlist                     | USN-4             | Pre-process the dataset  | 10           | Low      | Roshini V<br>Sangamithra V<br>Shalini M<br><br>Sharath Srinivas D |
| Sprint-3 | Predict                       | USN-5             | KNN model building   | 20           | High     | Roshini V<br>Sangamithra V<br>Shalini M<br><br>Sharath Srinivas D |
| Sprint-4 | Preference                    | USN-6             | Decision Tree model building   | 20           | High     | Roshini V<br>Sangamithra V<br>Shalini M<br><br>Sharath Srinivas D |

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

### Velocity :

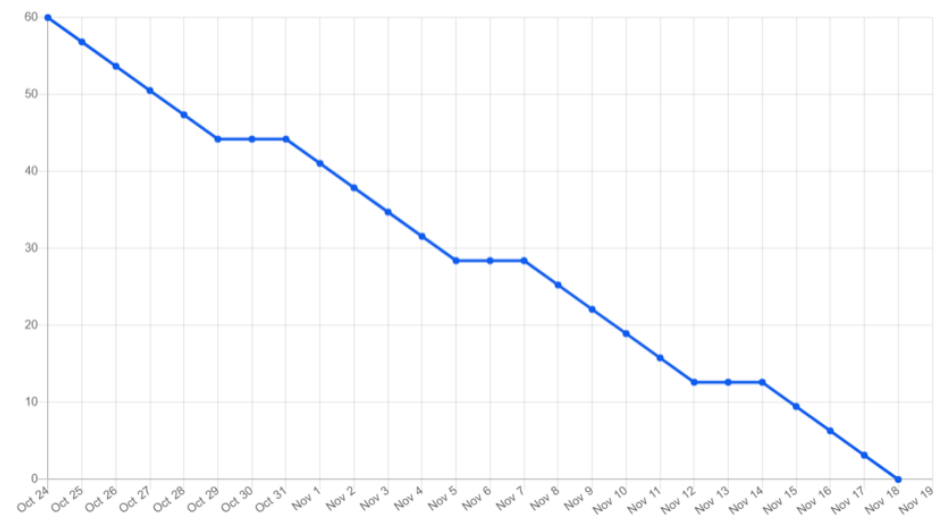
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$Av = 60 \text{ story points} / 24 \text{ days} = 2.5$$

### Burndown velocity :

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

Burndown Chart



## 7. CODING & SOLUTIONING

### 7.1 Feature 1

### University Admit Eligibility Predictor

GRE Score: 320  
TOEFL Score: 110  
University Rank:   
SOP:   
LOR:   
CGPA:   
Research:

### Predicting Chance of Admission

Prediction:  
Congrats! You're eligible

```

EXPLORER
  OPEN EDITORS
    app.py
  SPURIT 3
    static
      home.css
      money.jpeg
      rejected.jpeg
    templates
      approve.html
      home.html
      prediction.html
      reject.html
    app.py
    loan_prediction...
    model.pkl
    model.py

app.py
36 predict
37 proparea = request.form['proparea']
38 ApplicantIncome = float(request.form['ApplicantIncome'])
39 CopapplicantIncome = float(request.form['CopapplicantIncome'])
40 LoanAmount = float(request.form['LoanAmount'])
41 Loan_Amount_Term = float(request.form['Loan_Amount_Term'])
42 if gender == 'Male':
43     gender = 1
44 else:
45     gender = 0
46 if married == 'Yes':
47     married = 1
48 else:
49     married = 0
50 if education == 'Graduate':
51     education = 0
52 else:
53     education = 1
54 if employed == 'Yes':
55     employed = 1
56 else:
57     employed = 0
58 if dependents == '3+':
59     dependents = 3
60 if credit == 'Yes':
61     credit = 1
62 else:
63     credit = 0
64 if proparea == 'Urban':
65     proparea = 2
66 elif proparea == 'Rural':
67     proparea = 0
68 else:
69     proparea = 1
70 features = [gender, married, dependents, education, employed, ApplicantIncome,
71             CopapplicantIncome, LoanAmount, Loan_Amount_Term, credit, proparea]
72 con_features = [np.array(features)]
73 prediction = model.predict(con_features)
74 print(prediction)
75 if prediction == 1:
76     return render_template('approve.html', prediction_text='Congratulations! '+name+' You are eligible for loan')
77 else:
78     return render_template('reject.html', prediction_text='Sorry '+name+' You are not eligible for loan')
79
80 if __name__ == '__main__':
81     app.run(debug=True)

```

```

EXPLORER
  OPEN EDITORS
    app.py
  SPURIT 3
    static
      home.css
      money.jpeg
      rejected.jpeg
    templates
      approve.html
      home.html
      prediction.html
      reject.html
    app.py
    loan_prediction...
    model.pkl
    model.py

app.py
1 from flask import render_template, Flask, request
2 import numpy as np
3 import pickle
4 from sklearn.preprocessing import scale
5 app = Flask(__name__, template_folder='templates')
6
7 model = pickle.load(open("model.pkl", 'rb'))
8
9
10 @app.route('/')
11 def home():
12     return render_template('home.html')
13
14 @app.route('/login.html')
15 @app.route('/home.html')
16 def homeII():
17     return render_template('home.html')
18
19
20 @app.route('/prediction.html')
21 def forapp():
22     return render_template('prediction.html')
23
24
25 @app.route('/prediction.html', methods=['POST'])
26 def predict():
27     if request.method == 'POST':
28         name = request.form['Name']
29         gender = request.form['gender']
30         married = request.form['married']
31         dependents = request.form['dependents']
32         education = request.form['education']
33         employed = request.form['employed']
34         credit = request.form['credit']
35         proparea = request.form['proparea']
36         ApplicantIncome = float(request.form['ApplicantIncome'])
37         CopapplicantIncome = float(request.form['CopapplicantIncome'])
38         LoanAmount = float(request.form['LoanAmount'])
39         Loan_Amount_Term = float(request.form['Loan_Amount_Term'])
40         if gender == 'Male':
41             gender = 1
42         else:
43             gender = 0
44         if married == 'Yes':
45             married = 1
46         else:
47             married = 0
48         if education == 'Graduate':
49             education = 0
50         else:
51             education = 1
52         if employed == 'Yes':
53             employed = 1
54         else:
55             employed = 0

```

## 7.2 Feature 2

### Predicting Chance of Admission

Prediction:

Sorry! Better luck next time

[Back](#)

The image shows a web application running in a browser and its source code in VS Code. The browser window displays the title "Predicting Chance of Admission" and a message "Sorry! Better luck next time" with a "Back" link. The VS Code editor shows the source code for the application, which is a Flask web application. The code includes a Flask app object, a route for the prediction page, and a form for predicting the chance of admission. The form includes fields for Property Location, Applicant Income, Co-applicant Income, Purpose of loan, Loan Amount, Loan Amount Term, Aadhar Number, and PAN Card ID. The code also includes a function to predict the chance of admission based on the input data.

```
121 <option selected-- select Property Location --></option>
122 <option value="Semiurban">Semiurban</option>
123 <option value="Urban">Urban</option>
124 <option value="Rural">Rural</option>
125 </select>
126 </div>
127 <div class="mb-3">
128 <label for="exampleFormControlInput1" class="form-label">Enter Applicant Income</label>
129 <input type="text" class="form-control" id="ApplicantIncome" name="ApplicantIncome"
130 placeholder="Applicant Income" required>
131 </div>
132 <div class="mb-3">
133 <label for="exampleFormControlInput1" class="form-label">Enter Co-applicant Income</label>
134 <input type="text" class="form-control" id="CoapplicantIncome" name="CoapplicantIncome"
135 placeholder="Co-applicant Income" required>
136 </div>
137 <div class="mb-3">
138 <label for="exampleFormControlInput1" class="form-label">Purpose of loan</label>
139 <select class="form-select" id="pur" name="pur" aria-label="Default select example" required>
140 <option selected-- select the purpose of loan --></option>
141 <option value="Personal">Personal Loan</option>
142 <option value="Business">Business Loan</option>
143 <option value="Education">Education Loan</option>
144 <option value="Home">Home Loan</option>
145 <option value="Other">Other</option>
146 </select>
147 </div>
148 <div class="mb-3">
149 <label for="exampleFormControlInput1" class="form-label">Enter Loan Amount</label>
150 <input type="text" class="form-control" id="LoanAmount" name="LoanAmount" placeholder="Loan Amount" required>
151 </div>
152 <div class="mb-3">
153 <label for="exampleFormControlInput1" class="form-label">Enter Loan Amount Term</label>
154 <input type="text" class="form-control" id="Loan_Amount_Term" name="Loan_Amount_Term"
155 placeholder="Loan Amount Term" required>
156 </div>
157 <div class="mb-3">
158 <label for="exampleFormControlInput1" class="form-label">Enter Aadhar Number</label>
159 <input type="text" class="form-control" id="Aadhar" name="Aadhar" placeholder="Aadhar Number" required>
160 </div>
161 <div class="mb-3">
162 <label for="exampleFormControlInput1" class="form-label">Enter PAN Card ID</label>
163 <input type="text" class="form-control" id="PAN" name="PAN" placeholder="PAN Card ID" required>
164 </div>
165 <br>
166 <div class="mb-3">
167 <button type="submit" value="PREDICT" class="btn btn-dark">Predict</button>
168 </div>
169 </form>
170 </div>
```



### 7.3 Database Schema

The database used here in this project was Admission\_Predict.csv. The sample screenshot of the database are

481 lines (481 sloc) | 12.6 KB

Raw Blame

Search this file...

| 1  | Serial No. | GRE Score | TOEFL Score | University Rating | SOP | LOR | CGPA | Research | Chance of Admitt |
|----|------------|-----------|-------------|-------------------|-----|-----|------|----------|------------------|
| 2  | 1          | 337       | 118         | 4                 | 4.5 | 4.5 | 9.65 | 1        | 0.92             |
| 3  | 2          | 324       | 107         | 4                 | 4   | 4.5 | 8.87 | 1        | 0.76             |
| 4  | 3          | 316       | 104         | 3                 | 3   | 3.5 | 8    | 1        | 0.72             |
| 5  | 4          | 322       | 110         | 3                 | 3.5 | 2.5 | 8.67 | 1        | 0.8              |
| 6  | 5          | 314       | 103         | 2                 | 2   | 3   | 8.21 | 0        | 0.65             |
| 7  | 6          | 330       | 115         | 5                 | 4.5 | 3   | 9.34 | 1        | 0.9              |
| 8  | 7          | 321       | 109         | 3                 | 3   | 4   | 8.2  | 1        | 0.75             |
| 9  | 8          | 308       | 101         | 2                 | 3   | 4   | 7.9  | 0        | 0.68             |
| 10 | 9          | 302       | 102         | 1                 | 2   | 1.5 | 8    | 0        | 0.5              |
| 11 | 10         | 323       | 108         | 3                 | 3.5 | 3   | 8.6  | 0        | 0.45             |
| 12 | 11         | 325       | 106         | 3                 | 3.5 | 4   | 8.4  | 1        | 0.52             |
| 13 | 12         | 327       | 111         | 4                 | 4   | 4.5 | 9    | 1        | 0.84             |
| 14 | 13         | 328       | 112         | 4                 | 4   | 4.5 | 9.1  | 1        | 0.78             |
| 15 | 14         | 307       | 109         | 3                 | 4   | 3   | 8    | 1        | 0.62             |
| 16 | 15         | 311       | 104         | 3                 | 3.5 | 2   | 8.2  | 1        | 0.61             |
| 17 | 16         | 314       | 105         | 3                 | 3.5 | 2.5 | 8.3  | 0        | 0.54             |
| 18 | 17         | 317       | 107         | 3                 | 4   | 3   | 8.7  | 0        | 0.66             |
| 19 | 18         | 319       | 106         | 3                 | 4   | 3   | 8    | 1        | 0.65             |
| 20 | 19         | 318       | 110         | 3                 | 4   | 3   | 8.8  | 0        | 0.63             |
| 21 | 20         | 303       | 102         | 3                 | 3.5 | 3   | 8.5  | 0        | 0.62             |
| 22 | 21         | 312       | 107         | 3                 | 3   | 2   | 7.9  | 1        | 0.64             |
| 23 | 22         | 325       | 114         | 4                 | 3   | 2   | 8.4  | 0        | 0.7              |
| 24 | 23         | 328       | 116         | 5                 | 5   | 5   | 9.5  | 1        | 0.94             |
| 25 | 24         | 334       | 119         | 5                 | 5   | 4.5 | 9.7  | 1        | 0.95             |
| 26 | 25         | 336       | 119         | 5                 | 4   | 3.5 | 9.8  | 1        | 0.97             |

## 8. TESTING

### 8.1 Test Cases

| Section             | Total Cases | Not Tested | Fail | Pass |
|---------------------|-------------|------------|------|------|
| Print Engine        | 7           | 0          | 0    | 7    |
| Client Application  | 51          | 0          | 0    | 51   |
| Security            | 2           | 0          | 0    | 2    |
| Outsource Shipping  | 4           | 0          | 0    | 3    |
| Exception Reporting | 9           | 0          | 0    | 9    |
| Final Report Output | 4           | 0          | 0    | 4    |
| Version Control     | 2           | 0          | 0    | 2    |

## 8.2 User Acceptance Testing

### 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the University Admit Eligibility Predictor project at the time of the release to User Acceptance Testing (UAT).

### 2. Defect Analysis

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

| Resolution     | Severity<br>1 | Severity<br>2 | Severity<br>3 | Severity<br>4 | Subtotal |
|----------------|---------------|---------------|---------------|---------------|----------|
| By Design      | 10            | 4             | 2             | 3             | 19       |
| Duplicate      | 0             | 0             | 0             | 0             | 0        |
| External       | 2             | 3             | 0             | 1             | 6        |
| Fixed          | 11            | 2             | 4             | 20            | 37       |
| Not Reproduced | 0             | 0             | 0             | 0             | 0        |
| Skipped        | 0             | 0             | 1             | 1             | 2        |
| Won't Fix      | 0             | 0             | 0             | 0             | 0        |
| Totals         | 24            | 14            | 13            | 26            | 64       |

### 1. Test Case Analysis

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

| Section             | Total Cases | Not Tested | Fail | Pass |
|---------------------|-------------|------------|------|------|
| Print Engine        | 7           | 0          | 0    | 7    |
| Client Application  | 51          | 0          | 0    | 51   |
| Security            | 2           | 0          | 0    | 2    |
| Outsource Shipping  | 3           | 0          | 0    | 3    |
| Exception Reporting | 9           | 0          | 0    | 9    |
| Final Report Output | 4           | 0          | 0    | 4    |
| Version Control     | 2           | 0          | 0    | 2    |

## 9. RESULTS

### 9.1 Performance Metrics

#### Model Performance Testing:

Measure the performance using Metrics

```
pd.crosstab(Y_Test,y_predict)
```

| col_0 | 0   | 1   | 2   |
|-------|-----|-----|-----|
| Sex   |     |     |     |
| 0     | 108 | 29  | 112 |
| 1     | 33  | 223 | 35  |
| 2     | 123 | 52  | 121 |

```
print(classification_report(Y_Test,y_predict))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.41      | 0.43   | 0.42     | 249     |
| 1            | 0.73      | 0.77   | 0.75     | 291     |
| 2            | 0.45      | 0.41   | 0.43     | 296     |
| accuracy     |           |        | 0.54     | 836     |
| macro avg    | 0.53      | 0.54   | 0.53     | 836     |
| weighted avg | 0.54      | 0.54   | 0.54     | 836     |

#### Measuring the performance using metrics

```
from sklearn.metrics import mean_squared_error,mean_absolute_error
from sklearn.metrics import accuracy_score
mse = mean_squared_error(pred_test,y_test)
print("The Mean squared error is: ", mse)
rmse = np.sqrt(mse)
print("The Root mean squared error is: ", rmse)
mae = mean_absolute_error(pred_test,y_test)
print("The Mean absolute error is: ", mae)
acc = lr.score(x_test,y_test)
print("The accuracy is: ", acc)
```

```
The Mean squared error is: 3.403389401193475
The Root mean squared error is: 1.8448277429596172
The Mean absolute error is: 1.3537325298790688
The accuracy is: 0.0657871258637811
```

## 10. ADVANTAGES & DISADVANTAGES

Advantage:

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

Disadvantage:

1. Requires an active internet connection.
2. System will provide inaccurate results if data entered is incorrect.

## **11. CONCLUSION**

This system, which we constructed for the first time in Python employing ML algorithms and other front end languages like html, css, and java script, has proven to be more challenging than anticipated.

While it can seem straightforward to complete a few forms and process the data, there is much more that goes into choosing applications than this. Every time improvements and new features were implemented, ideas for more features or ways to make the system easier to use became obvious. A project in and of itself, balancing the completion of these required features with the suggestions for improvement as well as remembering everything that had to be done, was made possible by the fact that adding one feature made it possible to add another necessary feature. Finding out what needs to be debugged can often be a very simple task compared to debugging itself. Since so many components of the admissions system are interconnected, if an error is found on one page, it may be a display error, a problem with how the information is read from the database, or even a problem with how the information was initially stored in the database. Each time, all three must be verified. When the apparent reason for an issue is not immediately clear, this slows down the process and can be frustrating. The language used must be straightforward and simple to comprehend, and compatibility is crucial. The mobility of this system would not have been able to duplicate if it had not been created as a fully web-based application. Even if it doesn't have all the features that would have been preferred, the system performs well overall and meets initial expectations. The few mistakes that do occur are small or graphical, and the majority of the functionality that is provided works beautifully.

## **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 examining. 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.

## 13. APPENDIX

### Source Code

#### index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/styles.css">
  <title>University Admit Eligibility Predictor</title>
</head>
<body class="index-page"
  background-image: url(/home/sangamithra/Downloads/bg.jpeg);>
  <div class="index">
    <h1>University Admit Eligibility Predictor</h1>
    <a href="/checkEligibility"><button type="button" class="btns">Check
      Eligibility</button></a>
  </div>
</body>
</html>
```

#### Chance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/css/styles.css">
  <title>University Admit Eligibility Predictor</title>
</head>
<body class="predict">
  <div class="prediction-box">
    <h1>Predicting Chance of Admission</h1>
    <div class="prediction">
      <span>Prediction:</span>
```

```
        <h3 style="font-weight: 600;">You have a chance</h3>
    </div>
    <a href="/checkEligibility"><button type="button" class="btns">Back</button></a>
</div>
</body>
</html>
```

### Nochance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/css/styles.css">
    <title>University Admit Eligibility Predictor</title>
</head>
<body class="predict">
    <div class="prediction-box">
        <h1>Predicting Chance of Admission</h1>
        <div class="prediction">
            <span>Prediction: </span>
            <h3 style="font-weight: 600;">You don't have a chance</h3>
        </div>
        <a href="/checkEligibility"><button type="button" class="btns">Back</button></a>
    </div>
</body>
</html>
```

### Demo2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<link rel="stylesheet" href="/styles.css">
<title>University Admit Eligibility Predictor</title>
</head>
<body class="home">
  <h1>University Admit Eligibility Predictor</h1>
  <form method="post" action="/predict" class="form">
    <div class="form-data">
      <label for="greScore">GRE Score:</label>
      <input name="greScore" placeholder="GRE Score" required>
    </div>
    <div class="form-data">
      <label for="toeflScore">TOEFL Score:</label>
      <input name="toeflScore" placeholder="TOEFL Score" required>
    </div>
    <div class="form-data">
      <label for="univRank">University Rank:</label>
      <input name="univRank" placeholder="University Rank" required>
    </div>
    <div class="form-data">
      <label for="sop">SOP:</label>
      <input name="sop" placeholder="SOP" required>
    </div>
    <div class="form-data">
      <label for="lor">LOR:</label>
      <input name="lor" placeholder="LOR" required>
    </div>
    <div class="form-data">
      <label for="cgpa">CGPA:</label>
      <input name="cgpa" placeholder="CGPA" required>
    </div>
    <div class="form-data">
      <label for="research">Research:</label>
      <input name="research" placeholder="Research" required>
    </div>
    <br>
    <br>
    <button type="submit" class="btns">Submit</button>
```

```
<a href="/"><button type="button" class="btns">Back</button></a>
</form>
</body>
</html>
```

## **Model Training**

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

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

```
data.head()
```

```
data.tail()
```

```
data.drop(["Serial No."],axis=1,inplace=True)
data.head()
```

```
data.info()
```

```
data.describe()
```

```
"""## Data visualization"""
```

```
plt.scatter(data['CGPA'],data['Chance of Admit '])
plt.show()
```

```
plt.bar(data['GRE Score'],data['Chance of Admit '])
plt.show()
```

```
"""## Splitting the dependent and independent column"""
```

```
x=data.iloc[:,0:7].values
x.shape
```



```
y=data.iloc[:,7:].values
```

```
y.shape
```

```
"""## Splitting the data into train and test"""
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=10)
```

```
y_train=(y_train>0.5)
```

```
y_train
```

```
y_test=(y_test>0.5)
```

```
y_test
```

```
"""## Training and testing the model"""
```

```
X=data.drop(['Chance of Admit '],axis=1) #input data_set
```

```
y=data['Chance of Admit '] #output labels
```

```
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)
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler=MinMaxScaler()
```

```
X_train[X_train.columns] = scaler.fit_transform(X_train[X_train.columns])
```

```
X_test[X_test.columns] = scaler.transform(X_test[X_test.columns])
```

```
X_train.head()
```

```
from sklearn.ensemble import GradientBoostingRegressor
```

```
rgr = GradientBoostingRegressor()
```

```
rgr.fit(X_train,y_train)
```

```
rgr.score(X_test,y_test)
```

```
y_predict=rgr.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)

from sklearn.linear_model._logistic import LogisticRegression

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

lr = lore.fit(X_train, y_train)

y_pred = lr.predict(X_test)

from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix

print('Accuracy Score:', accuracy_score(y_test, y_pred))
print('Recall Score:', recall_score(y_test, y_pred))
print('ROC AUC Score:', roc_auc_score(y_test, y_pred))
print('Confusion Matrix:\n', confusion_matrix(y_test, y_pred))

"""## Save the model """

import pickle
pickle.dump(lr, open('university_admission.pkl', 'wb'))

model = pickle.load(open('university_admission.pkl', 'rb'))

```

### **Model Evaluation**

```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

```

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

```
data.head()
```

```
"""## Train and Testing the model"""
```

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

```
y=data['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
```

```
y_train
```

```
x_test
```

```
y_test
```

```
"""## Model Evaluation"""
```

```
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)
```

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

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

```
lr = lore.fit(x_train, y_train)
```

```
y_pred = lr.predict(x_test)
```

```
y_pred
```

## **13.2 GitHub& Project Demo Link**

### **GitHub link:**

<https://github.com/IBM-EPBL/IBM-Project-9130-1658982501>

### **Project Demo Link:**

<https://drive.google.com/file/d/1BWRDi-2O1ZJaz1-k-oJoUdI7TkYpX2PK/view>