

UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

A PROJECT REPORT

Submitted by

JASWIN J (962819104043)

AKASH STEPHEN C (962819104009)

ALDO SUJIN G (962819104013)

DHINESH NEELA THIRAVIAM P (962819104031)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

ANNA UNIVERSITY::CHENNAI 600 025

NOVEMBER 2022

INDEX

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 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

5.2.1. Technical Architecture

5.2.2. Technology Stack

5.3 User Stories

5.3.1. Customer Experience Journey Map

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.2.1. Velocity

6.3 Reports from JIRA

6.3.1. Burndown chart

7. CODING & SOLUTIONING

7.1 index.html

7.2 chance.html

7.3 nochance.html

7.4 style.CSS

7.5 app.py

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

8.2.1. Purpose of the Document

8.2.2. Defect Analysis

8.2.3. Test Case Analysis

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Student admission problem is very important in educational institutions. The world markets are developing rapidly and continuously looking for the best knowledge and experience among people. Young workers who want to stand out in their jobs are always looking for degrees in reputed universities that can help them in improving their skills and knowledge. This fact has motivated us to study the grades of students and the possibility of admission to graduate programs that can help the students who want to select a university for their higher studies in predicting the possibility of accepting the students each year and providing the needed resources. This project addresses machine learning models to predict the chance of a student to be admitted to a master's program. This will assist students to know in advance if they have a chance to get accepted. The machine learning models are multiple linear regression, k-nearest neighbor, random forest, and Multi-layer Perceptron. Experiments show that the Multi-layer Perceptron model surpasses other models. This can easily reduce the work of the student in manually analyzing the result and waiting for admission. The student can instantly get the result that he/she can get admitted in the selected university.

1.2 Purpose

The purpose of the project is to help students who want to select a university for their higher studies. The students may not know whether he/she get admitted into the university. To make this simple and reduce the work of the students before the admission process, this project helps the student whether he/she has the chance to get admitted into the university or not by analyzing various exam scores, thereby reducing the manual work. Thus the student can apply for the university where he/she has a good chance of getting admitted.

CHAPTER 2

LITERATURE SURVEY

1. The purpose of the literature survey is to give the brief overview and to establish complete information about the reference papers. The goal of literature survey is complete specify the technical details related to the main project in a concise and unambiguous manner.

Sushruta Mishra and Soumya Sahoo “A Quality Based Automated Admission System for Educational Domain”, pp.221-233, 2016.

2. In last two decades several educational institutes have started gaining momentum while many of them are in self-financing mode. Every institute wants to have good student strength to allow a smooth academic session. This paper proposes the use of machine learning techniques in educational domain to enhance the quality of student admissions in any higher educational institute. The focus of this paper is to identify those admissions inquires which most likely to turn into actual admissions. The result of analysis will assist the academic to take admission in the institution after enquiry.

Annam Mallikharjuna Roa, Nagineni Dharani, A. Satya Raghava, J. Buvanambigai, K. Sathish “College Admission Predictor” Vol. 8, Issue 4, 2018.

3. College Admission Predictor System is a web based application system in which students can register their marks along with their personal information. This helps to predict their admissions in colleges. Administrator can add the college details and the batch details. Using this Application, the entrance seat allotment becomes easier and efficient. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power for the allotment. Admin can add the allotted seats into a file and the details are

saved into the system. The total time for the entrance allotment becomes lower and the allotment process becomes faster. It helps students to make right decisions for choosing their college. In which students can register with their personal as well as marks details to prediction admission in colleges and the administrator can allot the seats for the students. Administrator can add the college details and the batch details. Using this Application, the entrance seat allotment becomes easier and efficient. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power for the allotment. Admin can add the allotted seats into a file and the details are saved into the system. The total time for the entrance allotment becomes lower and the allotment process becomes faster. It helps students to make right decisions for choosing their college.

Kiran Kumari, Meet Kataria, Viral Limbani, Rahul Soni “College Admission Predictor and Smart List Predictor Generator” 2019.

4. Students face a lot of difficulties to secure an admission in the college of their choice. The current scenario of an engineering admission process is little complicated and not so easy in terms of selecting an appropriate college according to the scores and field of interest. Accurate choice, varying with the entrance exam result and academic scores, is very important to the candidates to fill in the application form. There are many colleges offering multiple engineering courses. So it becomes trouble for some students to organize and list-out the proper colleges of their choice for courses according to their performance score. The CAPSLG system consists of a smart list generator working together with the help of college predictor, to aid students in the admission process. The college admission predictor uses historical colleges cut-off student admission data for predicting the most probable colleges. The system analyses student academic merits, background, and college admission criteria. Based on that, it predicts the likelihood of a university college that a student may enter. The smart list generator would enable the student to prepare the list of colleges, which could be needed to be filled in during the admission process. The system would

also get feedback from the users, which would prove helpful for prediction evaluation and improving the performance factor.

Md.Protikuzzaman, Mrinal Kanti Baowaly, Maloy Kumar Devnath and Bikash Chandra Singh “Predicting Undergraduate Admission”, Vol. 11, No. 12, 2020.

5. The University admission tests find the applicant’s ability to admit to the desired university. Nowadays, there is a huge competition in the university admission tests. The failure in the admission tests makes an examine depressed. This paper proposes a method that predicts undergraduate admission in universities. It can help students to improve their preparation to get a chance at their desired university. Many factors are responsible for the failure or success in an admission test. Educational data mining helps us to analyze and extract information from these factors. Here, the authors apply three machine learning algorithms XGBoost, LightGBM, and GBM on a collected dataset to estimate the probability of getting admission to the university after attending or before attending the admission test. They also evaluate and compare the performance levels of these three algorithms based on different evaluation metrics-accuracy and F1 score. Furthermore, the authors explore the important factors which influence predicting undergraduate admission.

Prince Golden, Kasturi Mojesh, Lakshmi Devarapalli, Pabbidi Naga Suba Reddy, Srigiri Rajesh, Ankita Chawla “A Comparative Study on Admission Predictions Using Machine Learning Techniques” Vol. 7, Issue 2, 2021.

2.1 Existing problem

Today the details of the student are manually entered in the records and details like merit list are prepared based on the score details. For this, the student must get the application form from the institution and fill out the details. Then with the details filled in the form, the admission process takes place by releasing the merit score and admission ID for

the candidate. This requires a lot of manpower and is very time-consuming. It takes much time for the student to get the result and make his/her choice of selecting the university.

2.2 References

Sushruta Mishra and Soumya Sahoo "A Quality Based Automated Admission System for Educational Domain", pp.221-233, 2016.

Annam Mallikharjuna Roa, Nagineni Dharani, A. Satya Raghava, J. Buvanambigai, K. Sathish "College Admission Predictor" Vol. 8, Issue 4, 2018.

Kiran Kumari, Meet Kataria, Viral Limbani, Rahul Soni "College Admission Predictor and Smart List Predictor Generator" 2019.

Md.Protikuzzaman, Mrinal Kanti Baowaly, Maloy Kumar Devnath and Bikash Chandra Singh "Predicting Undergraduate Admission", Vol. 11, No. 12, 2020.

Prince Golden, Kasturi Mojesh, Lakshmi Devarapalli, Pabbidi Naga Suba Reddy, Srigiri Rajesh, Ankita Chawla "A Comparative Study on Admission Predictions Using Machine Learning Techniques" Vol. 7, Issue 2, 2021.

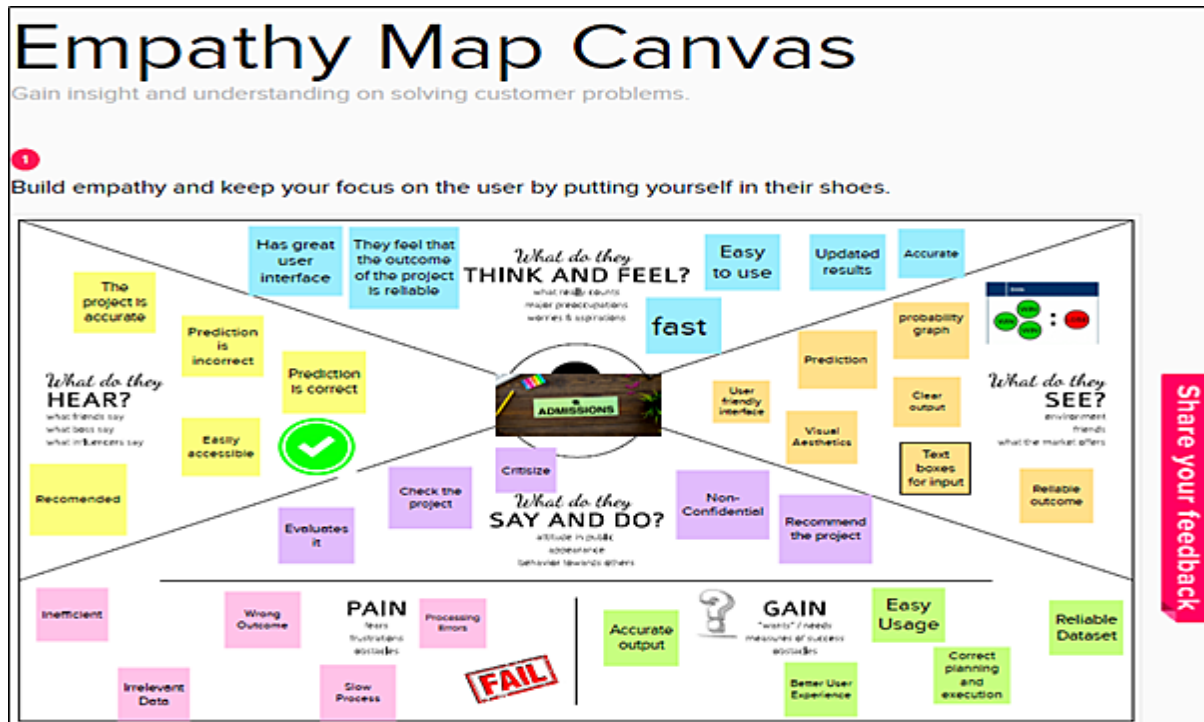
2.3 Problem Statement Definition

The problem includes the questions like whether a student will get an admit or not. What are the parameters for solutions in the university? What is the probability of being selected by the university? And can it be mathematically expressed? In this case, our objective is to predict whether a student will get an admit or not and the probability of getting admitted based on selection criteria.

CHAPTER 3

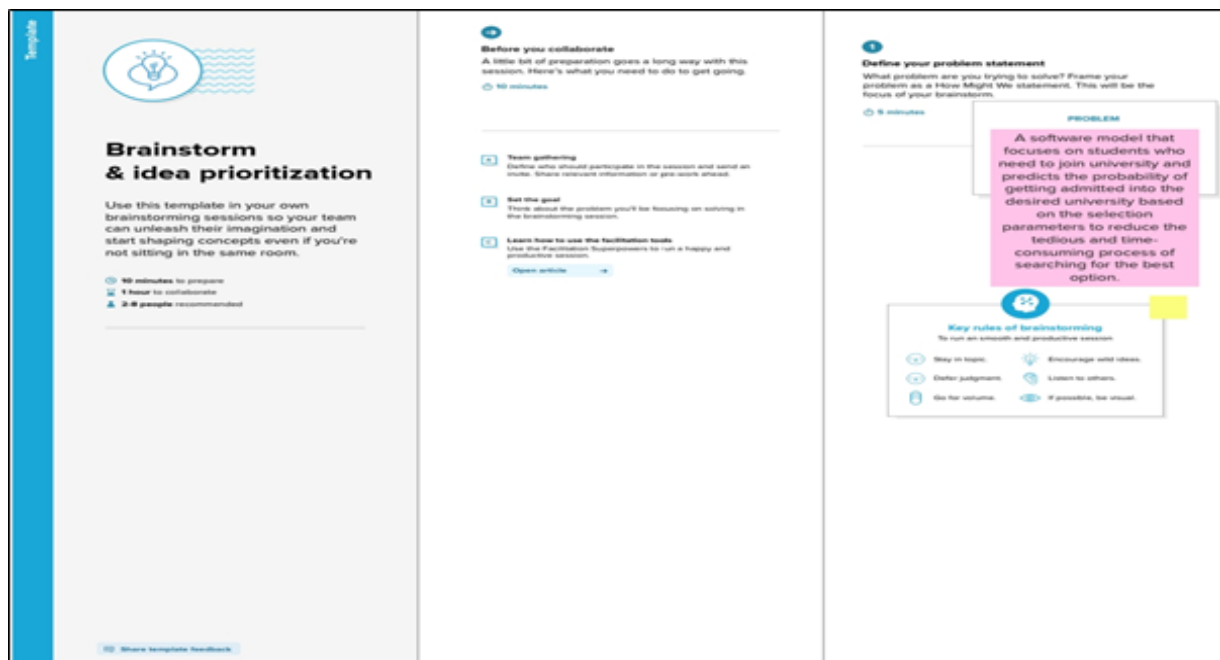
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

i) Defining the Problem Statement



ii) Brainstorming the Ideas

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!

Jaswin J

- Collect student details manually
- Store data in a database
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student
- Ask for details of university

Aldo Sujin G

- Collect student details manually
- Store data in a database
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student
- Ask for details of university

Akash Stephen C

- Collect student details manually
- Store data in a database
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student
- Ask for details of university

Dinesh Neela Thiraviyam P

- Collect student details manually
- Store data in a database
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student
- Ask for details of university

Person 5

-
-
-

Person 6

-
-
-

Person 7

-
-
-

Person 8

-
-
-

iii) Grouping the ideas

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 and 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.

INPUT DATA COLLECTION

- Collect student details manually
- Store data in a database
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student
- Ask for details of university

CHOOSE SELECTION

- Ask for details of university
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student

PERSON FROM METHOD

- Ask for details of university
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student

OUTPUT

- Ask for details of university
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student

DATASET ACCESS

- Collect student details manually
- Store data in a database
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student
- Ask for details of university

DATA COMMUNICATION

- Ask for details of university
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student

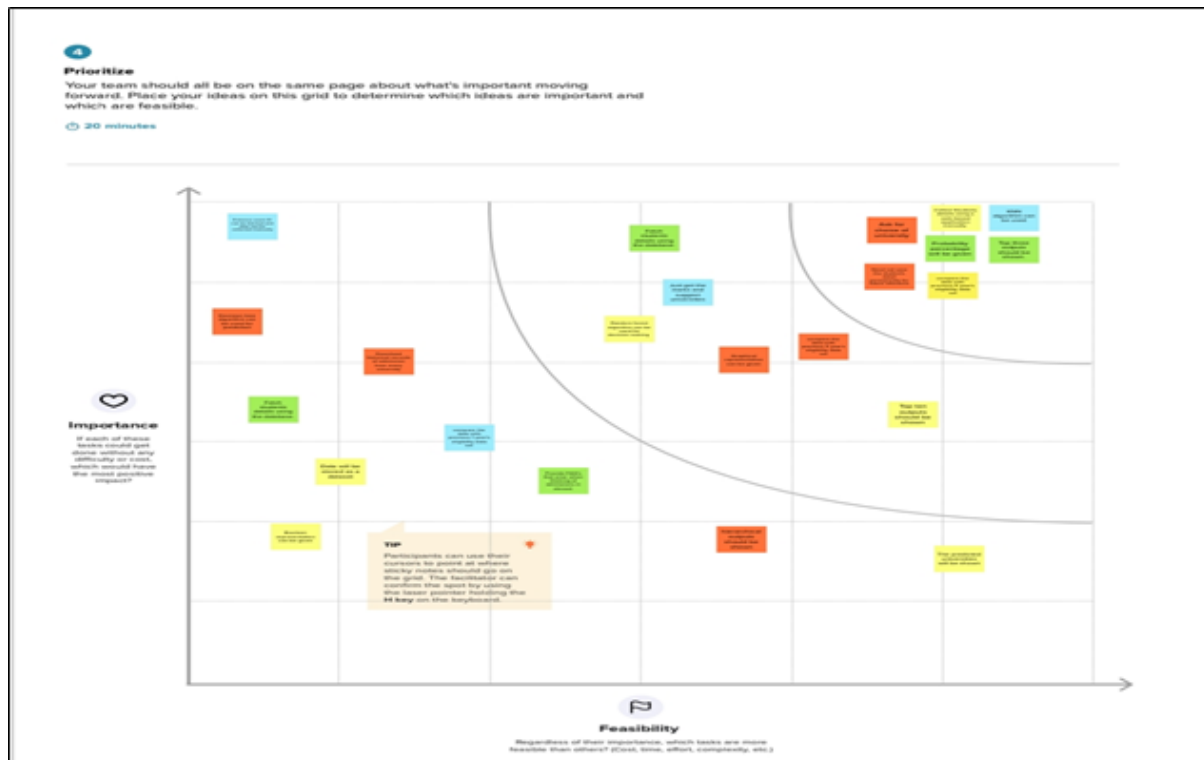
CHOOSE FACILITIES

- Ask for details of university
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student

OUTPUT RECOMMENDATION

- Ask for details of university
- Generate reports for the admin
- Implement a login system for the admin
- Implement a login system for the student

iv) Idea Prioritization



3.3 Proposed Solution

| S.No. | Parameter | Description |
|-------|---|---|
| 1. | Problem Statement (Problem to be solved) | The problem includes the questions like whether a student will get an admit or not. What are the parameters for admission to the university? What is the probability of being selected by the university? And can it be mathematically expressed? In this case, our objective is to predict whether a student will get an admit or not and the probability of getting admitted based on selection criteria. |
| 2. | Idea / Solution description | Collect Students' details using a web-based application manually. Then compare the data with the previous 5 year's eligibility data set. KNN algorithm can be used. The top three outputs should be shown. The probability percentage will also be displayed. |
| 3. | Novelty / Uniqueness | Based on the Student's detail the project shows the percentage of eligibility of the top universities preferred . |
| 4. | Social Impact / Customer Satisfaction | Prediction of the proposed system has a better accuracy rate and visual aesthetics that makes the user easy to understand the solution. This makes the customer satisfied. |
| 5. | Business Model (Revenue Model) | Promoting some universities and advertising them can provide us with |

| | | |
|----|-----------------------------|---|
| | | passive revenue. But the project is absolutely free for students' usage |
| 6. | Scalability of the Solution | More University Details can be updated which makes it easier for the user to browse details about much more universities. |

3.4 Problem Solution fit

| Project Title: University Admit Eligibility Predictor | | | Project Design Phase-I - Solution Fit | Template Team ID: PNT2022TMD34839 |
|---|--|--|--|--|
| 1. CUSTOMER SEGMENT(S) CS Who is your customer? I.e. working parents of 0-5 y.o. kids Students who have completed their schooling and searching for an eligible university do pursue their undergraduate program | | | 6. CUSTOMER CONSTRAINTS CC 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. Accuracy of the system, Doubts about prediction, Reliability, budget. | 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 rehearsal. The students can get the correct probability based on their eligibility criteria. If they are not eligible, the next best option is given as prescription. Manually checking the previous years' dataset is an alternative method. |
| 2. JOBS-TO-BE-DONE / PROBLEMS JLP Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides. Predicting the University that they are eligible to be admitted based on their eligibility criteria. Suggesting universities. Predicting the probability of their admission to the selected university based on the eligibility criteria. | | | 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. Students do not have time to manually analyse the course details of all universities and about the courses in which they have the area of interest. The availability of seats for the scores that they have got also can't easily be predicted. | 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) The student should give the details of their eligibility criteria correctly and when the predictions are given as output, they need to filter out their choices by selecting the right university he/she is eligible to be admitted. |

| | | | | |
|---|--|--|---|---|
| 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 who get to know about the platform from previous users who have got correct predictions. The Search for finding a platform in a browser to predict admit eligibility can also be a driving force. | | | 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 today. 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. This project gives the exact probability of getting admitted into the university that the student selects based on their scores and other criteria and also provides them with suggestions based on the same. This effectively decreases the time spent in finding a suitable university. | 8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. 8.1 ONLINE The student can check the university admission criteria online. 8.2 OFFLINE The student can personally visit the university in person and proceed with the admission process |
|---|--|--|---|---|

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|--------|-------------------------------|--|
| FR-1 | User Registration | Registration through Form Registration through Gmail Registration through LinkedIn |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | User Details | Submit the documents <ol style="list-style-type: none">1. GRE or/and TOEFL Score Sheet2. Curriculum Vitae (CV)3. Statement of purpose(SOP)4. Letter of Recommendation |
| FR-4 | User Requirements | <ol style="list-style-type: none">1. Upload all the relevant documents in the appropriate location in the website2. Based on the uploads the system would scrape all the necessary information3. The List of all possible University for the candidate would be displayed based on the scraped information |

4.2 Non-Functional requirements

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

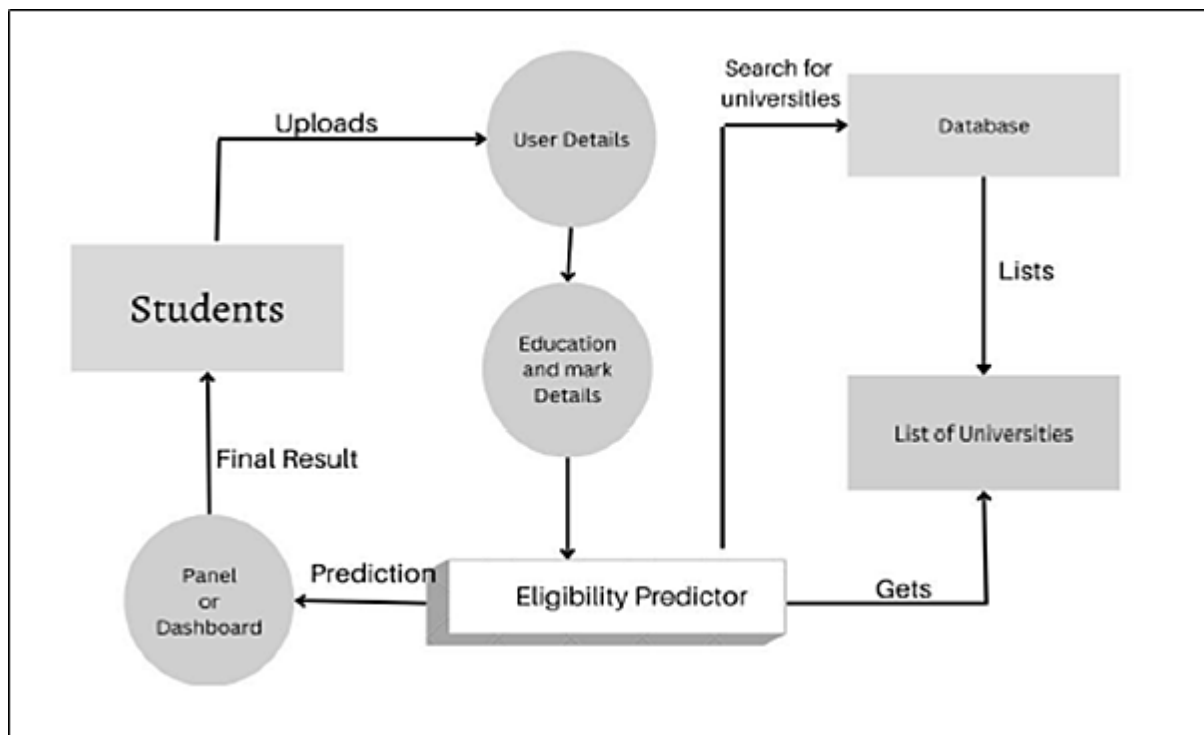
| NFR No. | Non-Functional Requirement | Description |
|----------------|-----------------------------------|--|
| NFR-1 | Usability | <ol style="list-style-type: none">1. Good User Interface.2. The interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand. |
| NFR-2 | Security | <ul style="list-style-type: none">➤ Data inside the system will be protected against malware attacks or unauthorized access. |
| NFR-3 | Reliability | <ul style="list-style-type: none">➤ The prediction made by the model is very accurate so that it is always reliable. |
| NFR-4 | Performance | <ul style="list-style-type: none">➤ Limited control over third party domains. |
| NFR-5 | Availability | <ul style="list-style-type: none">➤ As it is an online software model it is available 24/7 and can be used by many numbers of students at the same time. |
| NFR-6 | Scalability | <ul style="list-style-type: none">➤ Able to manage numerous concurrent users. |

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams

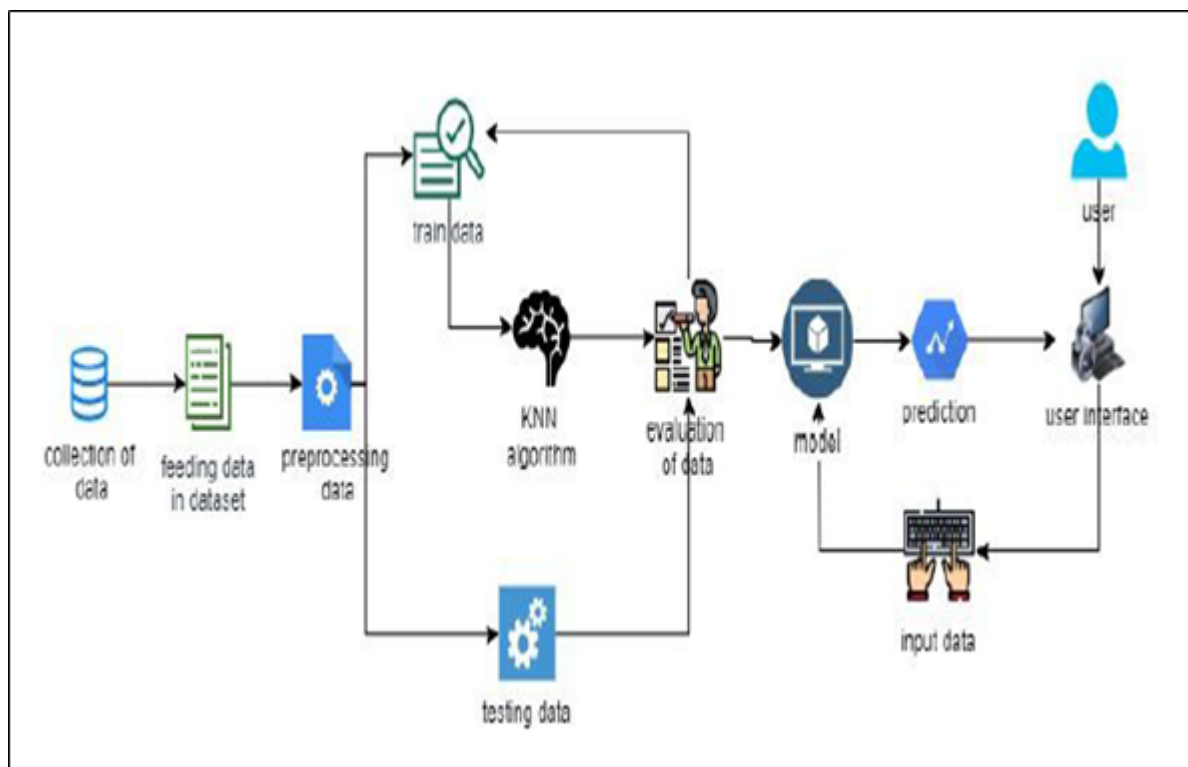
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

- First, the data set that contains the information about previous years' admission criteria was collected and fed in the system for preprocessing as a .CSV file.
- Then with the help of the data set, the model is trained and tested for expected input values by methods like KNN algorithm and regression techniques.
- Finally, with the help of the user interface, inputs from the user is obtained and prediction is done by the model and output is given.

5.2.1. Technical Architecture



5.2.2. Technology Stack

Table-1 : Components & Technologies:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|--|--|
| 1. | Open-Source Frameworks | Python is used for backend purpose and flask is imported for front end pupose | Python, Flask |
| 2. | Security Implementations | Data inside the system will be protected against malware attacks or unauthorized access | SHA-256, Encryptions, IAM Controls, OWASP etc. |
| 3. | Scalable Architecture | The accurate list of eligible universities name and user manual will be provided | KNN Algorithm |
| 4. | Availability | It is available 24/7 and anyone can use it at anytime | IBM Load Balancer |
| 5. | Performance | The user can have a knowledge of their eligibility for applying universities through our website | KNN Algorithm |

Table-2: Application Characteristics:

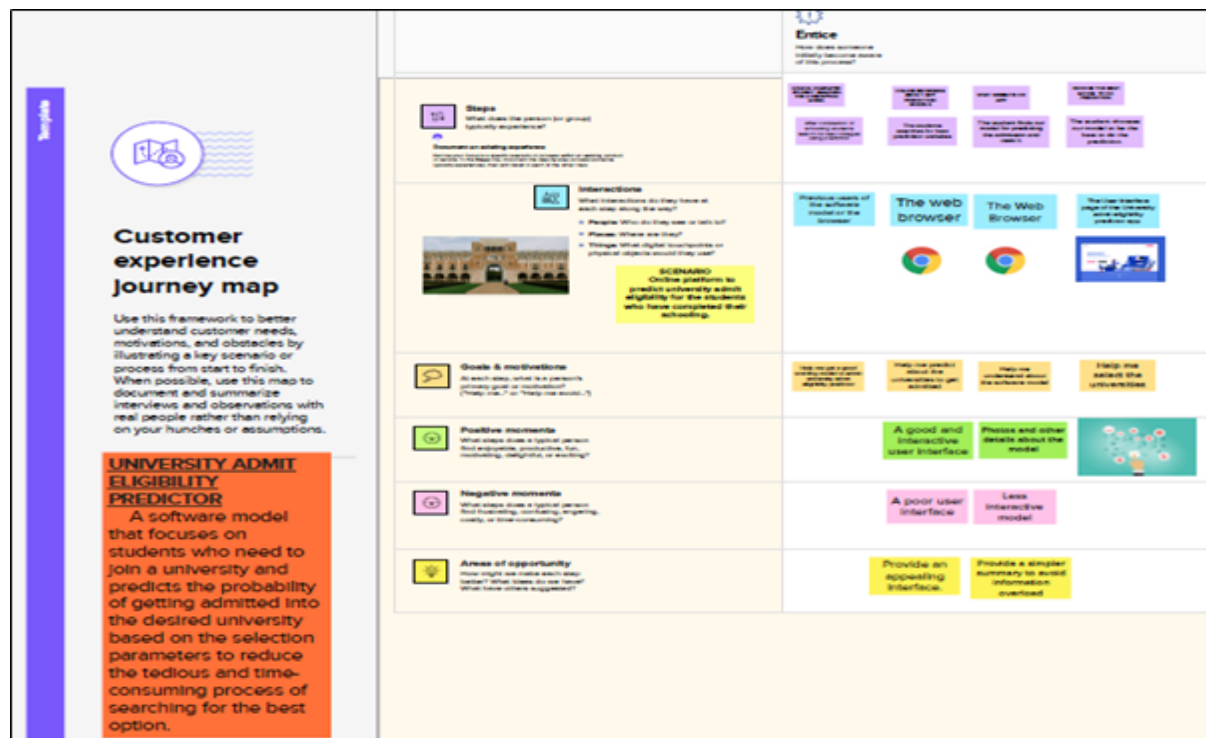
| S.No | Component | Description | Technology |
|------|---------------------------------|--|--|
| 1. | User Interface | User can interacts with the application through Web UI | HTML, CSS, JavaScript, Bootstrap, Flask |
| 2. | Application Logic-1 | Students can enter the required data in the form and it is displayed using flask and send to machine learning model for prediction | Python |
| 3. | Application Logic-2 | The application is directly deployed in the IBM cloud | IBM Watson STT service |
| 4. | Application Logic-3 | It uses AI to give fast, consistent and accurate answers for application | IBM Watson Assistant |
| 5. | Database | The user credentials is stored and used to send notification of any updates | MySQL |
| 6. | Cloud Database | Use to organize, store and manage data within the organisation. | IBM DB2, IBM Cloud |
| 7. | File Storage | Stores data in a hierarchical structure | IBM Block Storage or Other Storage Service or Local Filesystem |
| 8. | External API-1 | It checks the user data in accordance with the government standard and validate the user data provided | Aadhaar API |
| 9. | Machine Learning Model | The model is used to predict whether the student is eligible or not | Object Recognition Model, etc. |
| 10. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud System | Local, Cloud Foundry, Kubernetes, etc. |

5.3 User Stories

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---------------------------------|-------------------------------|-------------------|---|---|----------|----------|
| Customer (Mobile user/Web user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
| | | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
| | | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-2 |
| | | USN-4 | As a user, I can register for the application through Gmail | I can register & access the dashboard with Gmail Login | Medium | Sprint-1 |
| | Login | USN-5 | As a user, I can log into the application by entering email & password | I can access the dashboard | High | Sprint-1 |
| | Dashboard | USN-6 | As a user, I can view the details such as my profile and academic details | I can update my profile | Medium | Sprint-1 |
| Customer Care Executive | Support | USN-7 | As a customer care executive, responding to queries via phone calls, live chat etc. | Send immediate response | Medium | Sprint-3 |
| | | USN-8 | As a customer care executive, Ask for act on user feedback | Thank user for feedback | High | Sprint-2 |
| | | USN-9 | As a customer care executive, analyse customer data and communication to adjust customer care | Look into that issue soon and try to rectify it | Low | Sprint-3 |
| Administrator | Landing page | USN-10 | As an administrator, I shall update the news about university | Check the update is reflected or not | Medium | Sprint-4 |
| | Chances | USN-11 | As an administrator, I can send the prediction details to the users | I can provide chance for joining the university | Medium | Sprint-4 |
| | Update | USN-12 | As an admin, I can update the university database depends upon the user request | I can update university database | High | Sprint-4 |

5.3.1. Customer Experience Journey Map

- Project Statement and Entice Step



- Enter, Engage, Exit and Extend Steps

| Enter What is people experience as they begin the process? | Engage In the core moments in the process, what happens? | Exit What is people typically experience as the process finishes? | Extend What happens after the experience is over? |
|---|--|---|--|
| <p>Registration</p> <p>Students register using required personal and contact information</p> <p>After registering, students must fill in the required fields for prediction</p> <p>Students select the college they want to go to</p> | <p>After predicted output</p> <p>Students can see the results of the college prediction</p> <p>Students can see the results of the college prediction</p> <p>Students can see the results of the college prediction</p> | <p>Leave to Exit</p> <p>After getting the required prediction, students can leave the app</p> <p>Submit</p> <p>After getting the required prediction, students can leave the app</p> <p>Leave to Exit</p> <p>After getting the required prediction, students can leave the app</p> | <p>Feedback</p> <p>Students can provide feedback on the app</p> <p>Feedback</p> <p>Students can provide feedback on the app</p> <p>Feedback</p> <p>Students can provide feedback on the app</p> |
| <p>Register page of the University admission eligibility predictor app</p> <p>Enter the page of the University admission eligibility predictor app</p> <p>University selection page of the University admission eligibility predictor app</p> | <p>Result page of the University admission eligibility predictor app</p> <p>University selection page of the University admission eligibility predictor app</p> <p>Result page of the University admission eligibility predictor app</p> | <p>Exit page of the University admission eligibility predictor app</p> <p>Feedback page of the University admission eligibility predictor app</p> <p>Exit page of the University admission eligibility predictor app</p> | <p>App notifications</p> <p>App notifications</p> |
| <p>Help me register my details</p> <p>Help me fill my scores and required details</p> <p>Help me search and select the required college</p> | <p>Help me get the correct prediction about admission</p> | <p>Help me to leave the page</p> <p>Help me to provide feedback</p> <p>Help me to leave the app</p> | <p>Help me to get frequent updates</p> |
| <p>Well developed interface</p> <p>Detailed model</p> | <p>Predictions with good Pictorial / graphical representation</p> | | <p>Frequent updates and suggestions for the user</p> |
| <p>App with low level interface</p> | <p>Unsatisfying output format</p> <p>The prediction is not understandable</p> | <p>When there is no other facility in the app</p> | <p>When the predictions go wrong</p> |
| | <p>Provide a visually enhanced output</p> | <p>Provide users with other details</p> | <p>Updating users with useful notifications</p> |

CHAPTER 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 can register for the application by entering my email, password, and confirming my password. | 5 | High | 2 |
| Sprint-1 | Login | USN-2 | As a user, I can log into the application by entering email & password | 3 | High | 1 |
| Sprint-1 | Update Profile | USN-3 | As a user, after logging in, I will have to update my profile by providing all the required details. | 5 | Low | 2 |
| Sprint-2 | Choose University | USN-4 | As a user, I will be able to view the list of Universities that the students are eligible to apply. | 8 | Medium | 3 |
| Sprint-2 | Choose Course | USN-5 | As a user, I will be able to view the list of courses that the students are eligible to apply. | 8 | High | 3 |
| Sprint-2 | Admission Process | USN-6 | As a user, I will be able to view the details of Admission process like date and venue of certification verification. | 9 | High | 4 |
| Sprint-3 | Authentication | USN-7 | As an admin, the login credential of the user is authenticated by me. | 11 | High | 5 |
| Sprint-3 | Update Profile | USN-8 | As an admin, I can verify the user entered details. | 5 | Low | 2 |
| Sprint-4 | Prediction | USN-9 | As an admin, I can test the trained ML model by analysing the user details by ML algorithms like Logistic Regression. | 7 | Medium | 3 |
| Sprint-4 | Output | USN-10 | As an admin, I can upload the confirmation of user for the prediction into the Database. | 7 | Medium | 3 |

6.2 Sprint Delivery Schedule

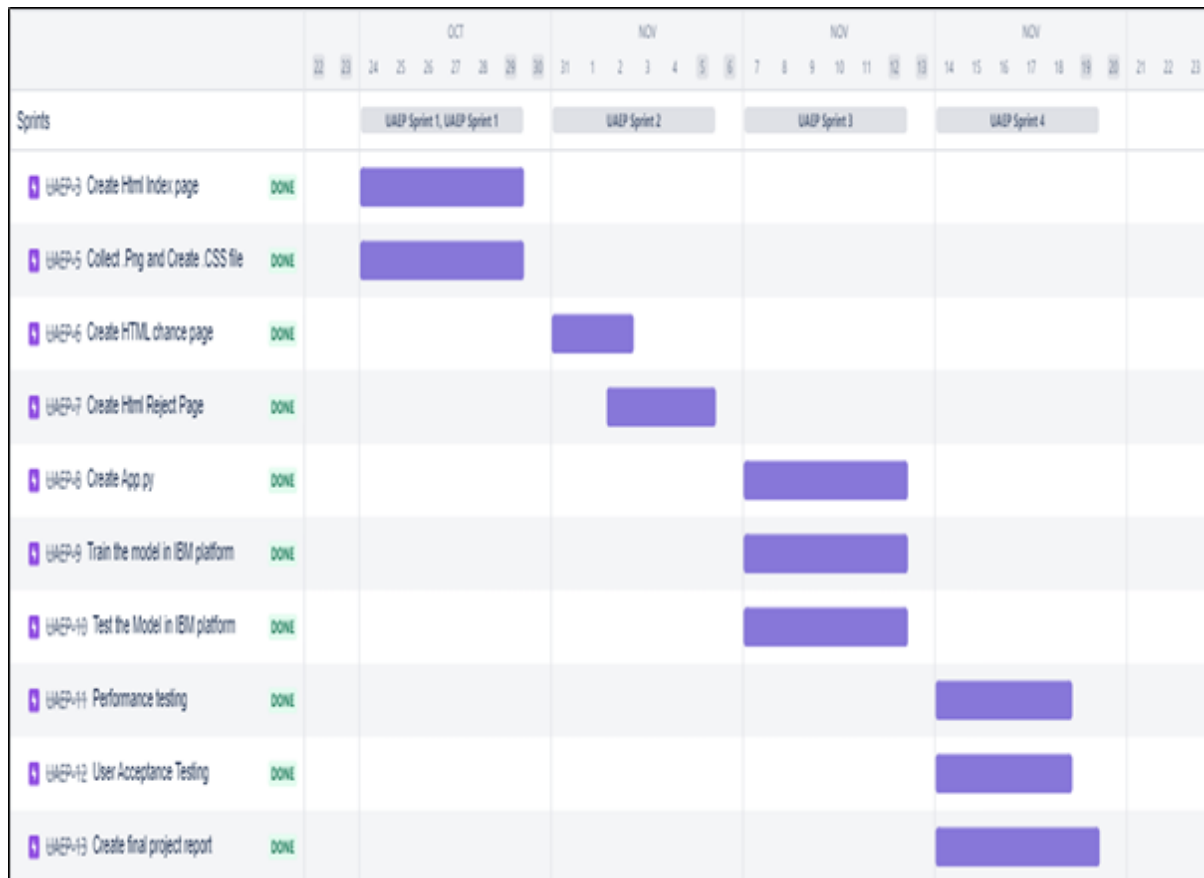
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|----------|--------------------|----------|-------------------|---------------------------|---|------------------------------|
| Sprint-1 | 13 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 13 | 29 Oct 2022 |
| Sprint-2 | 25 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 25 | 05 Nov 2022 |
| Sprint-3 | 16 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 16 | 12 Nov 2022 |
| Sprint-4 | 14 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 14 | 19 Nov 2022 |

6.2.1. Velocity

$$\text{Average Velocity (AV)} = \frac{\text{Sprint Duration}}{\text{Velocity}}$$

| SPRINT | STORY POINTS | DURATION | AVREAGE VELOCITY |
|----------|--------------|----------|------------------|
| Sprint-1 | 13 | 6 Days | = 2.167 |
| Sprint-2 | 25 | 6 Days | = 4.167 |
| Sprint-3 | 16 | 6 Days | = 2.67 |
| Sprint-4 | 14 | 6 Days | = 2.44 |
| OVERALL | 68 | 24 Days | = 2.83 |

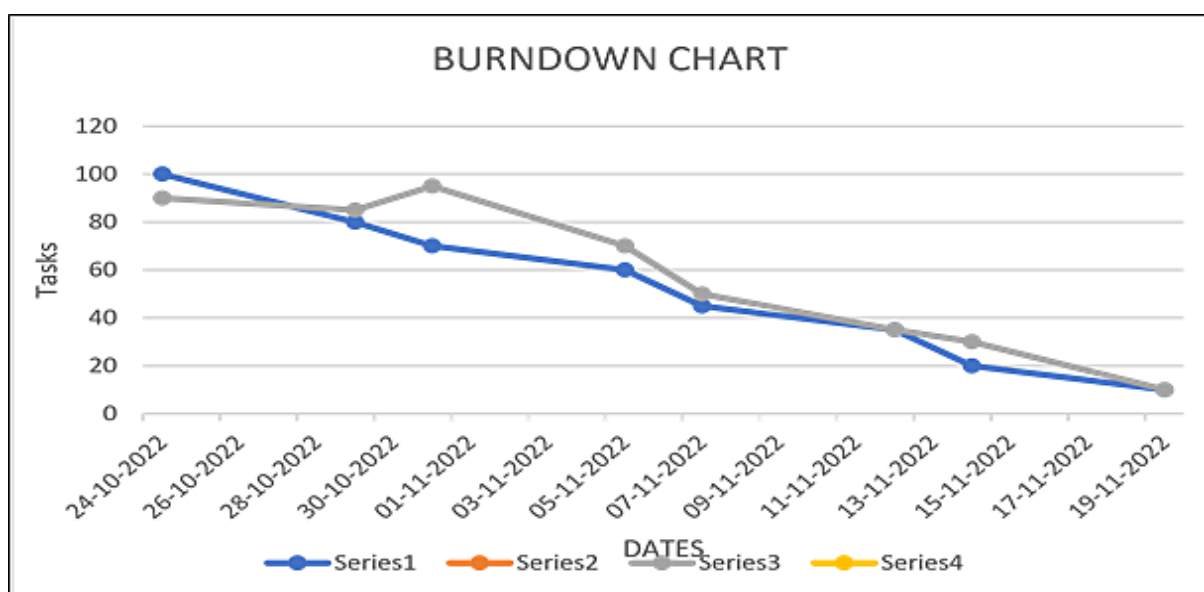
6.3 Reports from JIRA



6.3.1.Burndown chart:

A burn down 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 | | |
|----------------|---------------|--------------|
| DATES | PLANNED TASKS | ACTUAL TASKS |
| 24-10-2022 | 100 | 90 |
| 29-10-2022 | 80 | 85 |
| 31-10-2022 | 70 | 95 |
| 05-11-2022 | 60 | 70 |
| 07-11-2022 | 45 | 50 |
| 12-11-2022 | 35 | 35 |
| 14-11-2022 | 20 | 30 |
| 19-11-2022 | 10 | 10 |



CHAPTER 7

CODING & SOLUTIONING

7.1. 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="../static/css/style.css">
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.
min.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv
1WTRi" crossorigin="anonymous">
<title>University Eligibility Predictor</title>
</head>
<body>
<div id="body">
<h1 class="text-center mt-3">University Eligibility Predictor</h1>
<p class="text-center mt-3 font-weight-normal">This website is used to
check the Eligibility of a student to study in the top university</p>
<div class="heading">
<div class="ml-1 p-4">

<p class="mt-2 text-center">Statistics</p>
</div>
<div class="ml-1 p-4">

<p class="mt-2 text-center">Shortlist top university</p>
</div>
```

```

<div class="ml-2 p-4">

<p class="mt-2 text-center">Profile evaluated</p>
</div>
</div>
<div class="col-6 " id="main" >
<div class="card p-2 ms-2 my-2" style="background-color:
paleturquoise;">
<div class="card-body" >
<h5 class="card-title pb-4 text-center">Enter the details </h5>
<form action="/" method="post" style="background-color:lightskyblue ;"
id="theForm">
<div class="row mb-3">
<label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
<div class="col-lg-10">
<input type="number" class="form-control" id="gre" name="gre"
min="250" max="340" placeholder="250 to 340" required>
</div>
</div>
<div class="row mb-3">
<label for="tofel" class="col-lg-2 col-form-label">TOFEL Score:</label>
<div class="col-lg-10">
<input type="number" class="form-control" id="tofel" name="tofel"
min="50" max="120" placeholder="50 to 120" required>
</div>
</div>
<div class="row mb-3">
<label for="university_rating" class="col-lg-2 col-form-label">University
Rating:</label>
<div class="col-lg-10">
<input type="number" class="form-control" id="university_rating"
step="0.01" name="university_rating" min="1" max="5" placeholder="1
to 5" required>

```

```

</div>
</div>
<div class="row mb-3">
<label for="sop" class="col-lg-2 col-form-label">SOP:</label>
<div class="col-lg-10">
<input type="number" class="form-control" id="sop" name="sop"
step="0.01" min="1" max="5" placeholder="1 to 5" required>
</div>
</div>
<div class="row mb-3">
<label for="lor" class="col-lg-2 col-form-label">LOR:</label>
<div class="col-lg-10">
<input type="number" class="form-control" id="lor" name="lor"
step="0.01" min="1" max="5" placeholder="1 to 5" required>
</div>
</div>
<div class="row mb-3">
<label for="cgpa" class="col-lg-2 col-form-label">CGPA:</label>
<div class="col-lg-10">
<input type="number" class="form-control" id="cgpa" name="cgpa"
step="0.01" min="5" max="10" placeholder="5 to 10" required>
</div>
</div>
<fieldset class="row mb-3">
<legend class="col-form-label col-sm-2 pt-0">Research:</legend>
<div class="col-sm-10">
<div class="form-check">
<input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios1" value="1">
<label class="form-check-label" for="yes_no_radio">Yes</label>
</div>
<div class="form-check">

```

```

<input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios2" value="0" checked>
<label class="form-check-label" for="yes_no_radio">No</label>
</div>
</div>
</fieldset>
<div class="row lg-3 justify-content-center">
<div class="col-lg-2 mb-2 me-3">
<button type="submit" class="btn btn-primary m-auto"
style="padding:8px 25px; " id="button">Predict</button>
</div>
</div>
</form>
</div>
</div>
</div>
</div>
</body>
</html>

```

7.2.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="../static/css/style.css">
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.
min.css" rel="stylesheet" integrity="sha384-

```

```

ZenH87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5lDxbcnCeuOxjzrPF/et3URy9Bv
1WTRi"      crossorigin="anonymous">
<title>Document</title>
</head>
<body>
<div id="body">
<h1 class="text-center" id="title" ><strong>Hello there let's see the
chance!</strong></h1>
<div class="m-5 d-flex flex-row ">
<div class="col-sm-8">
<div class="alert alert-success" style="height:70px;" id="box">
<strong>Success!</strong> There is a high percent chance for you in
this rating university.
</div>
<p style="font-size:22px; font-weight: 500;">You have a high chance of
getting an opportunity at this university.<ul style="font-size:22px;"><li>
You can apply for this rating universities</li><li>OR      You can try
higher-rating universities</li></ul></p>
</div>
<div class="m-5" id="mark2"></div>
</div>
<div class="m-5 d-flex flex-row ">
<div class="col-sm-8"><p class="text-center pt-5" style="font-size:33px;
font-weight: 500;">Good      Luck!!!</p></div><div class="">

</div>
</div>
<div class="d-flex justify-content-center">
<button class="btn btn-primary " style="margin-top: 0;"><a href="/"
style="color:aqua; text-decoration : none" >Go Back</a></button>

```

```
</div>
</div>
</body>
</html>
```

7.3.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="../static/css/style.css">
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.
min.css" rel="stylesheet"      integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv
1WTRi"      crossorigin="anonymous">
<title>Document</title>
</head>
<body>
<div id="body">
<h1 class="text-center" id="title" ><strong>Hello there let's see the
chance!</strong></h1>
<div class="m-5 d-flex flex-row ">
<div class="col-sm-8">
<div class="alert alert-warning" style="height:70px;" id="box1">
<strong>Success!</strong> There is a low percent chance for you in this
rating university.
</div><br>
```

```

<p style="font-size:22px; font-weight: 400;">You have a meager chance
of getting opportunity in this university.<br>so you can proceed with
lower-rating universities</p>
<strong style="font-size:30px;">To boost your chance:</strong>
<ul style="font-size:22px;">
<li>Re-take the exam if you take lower marks than the median <br>It will
boost your mark</li>
</ul>
</div>
<div class="m-5" id="image"></div>
</div>
<div class="m-5 d-flex flex-row ">
<div class="col-sm-8"><p class="text-center pt-5" style="font-size:33px;
font-weight: 500;">Don't Lose      Hope!!!</p></div>
<div class="">

</div>
</div>
<div class="d-flex justify-content-center">
<button class="btn btn-primary " style="margin-top: 0;"><a href="/"
style="color:white; text-decoration : none" >Go Back</a></button>
</div>
</div>
</body>
</html>

```


7.4.style.css

```
.heading{
    display: flex;
    justify-content: space-around;
    margin-top: 45px;
}
#resultImage{
    height: 280px;
    width: 280px;
}
#title{
    margin-top: 50px;
    font-style: italic;
    font-weight: 400;
    font-size: 50px;
}
#body{
    background-color: blue;
    background-size: 100%;
}
#image{
    height: 335px;
    width: 335px;
}
#box{
    background-color: aqua;
    background-size: 100%;
}
#box1{
```

```

        background-color: pink;
    }
    img{
        width:100px;
        height: 100px;
    }
    #main{
        margin:auto;
    }
    #mark1{
        font-size:110px;
        height: 250px;
        width: 250px;
        color: rgb(62, 180, 62)
    }
    #mark2{
        font-size:110px;
        height: 250px;
        width: 250px;
        color: red;
    }

```

7.5.app.py

```

from flask import Flask, render_template, redirect, url_for, request
import requests
app = Flask(__name__)
@app.route("/", methods=['POST', 'GET'])
def index():
    if request.method == 'POST':

```

```

arr = []
for i in request.form:
    val = request.form[i]
    if val == "":
        return redirect(url_for("demo2"))
    arr.append(float(val))

# deepcode ignore HardcodedNonCryptoSecret: <please specify a
reason of ignoring this>
API_KEY = "zdtjceMI7i_9it0_bSV4p_1xzLiZALY8DZKGVfNWmgDh"
token_response =
requests.post('https://iam.cloud.ibm.com/identity/token', data={
    "apikey": API_KEY,
    "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}
payload_scoring = {
    "input_data": [{"fields": ['GRE Score',
                                'TOEFL Score',
                                'University Rating',
                                'SOP',
                                'LOR ',
                                'CGPA',
                                'Research'],
                    "values": [arr]
                }]
}
response_scoring = requests.post(
    'https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/5f722bae-

```

```

6e06-4163-aa2c-7f6273b346ae/predictions?version=2022-11-18',
    json=payload_scoring,
    headers=header
).json()
result = response_scoring['predictions'][0]['values']
if result[0][0] > 0.5:
    return redirect(url_for('chance', percent=result[0][0] * 100))
else:
    return redirect(url_for('no_chance', percent=result[0][0] * 100))
else:
    return redirect(url_for("demo2"))
@app.route("/home")
def demo2():
    return render_template("index.html")
@app.route("/chance/<percent>")
def chance(percent):
    return render_template("chance.html", content=[percent])
@app.route("/nochance/<percent>")
def no_chance(percent):
    return render_template("noChance.html", content=[percent])
@app.route('/<path:path>')
def catch_all():
    return redirect(url_for("demo2"))
if __name__ == "__main__":
    app.run()

```

CHAPTER 8

TESTING

8.1 Test Cases

| | | | | Date | 15-Nov-22 | | | | | | |
|----------------|--------------|-------------|---|---------------|--|---------------------|---|---------------------|--------|--|---------------------------|
| | | | | Team ID | PNT2022TMD34899 | | | | | | |
| | | | | Project Name | University Admit Eligibility Predictor | | | | | | |
| | | | | Maximum Marks | 4 marks | | | | | | |
| Test case ID | Feature Type | Component | Test Scenario | Pre-Requisite | Steps To Execute | Test Data | Expected Result | Actual Result | Status | command | Executed By |
| Index_page_01 | Functional | Home Page | Verify user is able to see the fields for entering the scores | | 1. Enter URL and click go | Score Details | All the text fields should display | Working as expected | Pass | | Jarvin J |
| Index_page_02 | UI | Home Page | Verify the UI elements | Score Detail | 1. Enter URL and click go 2. Submit the score details | Score Details | Application should show below UI elements: A. Information about the project, B. Text boxes for filling details, C. Submit button | Working as expected | pass | | Dhinesh Neela Thiraviam P |
| Index_page_03 | Functional | Home page | Verify user is able to get error messages when given wrong details. | Score Detail | 1. Enter URL and click go 2. Submit the Wrong Score Details | Wrong Score Details | give wrong score details as input data | Working as expected | pass | An error pop-up message is shown to the user | Aldo Segin G |
| Result_page_01 | Functional | Result Page | Verify the UI elements | | 1. Enter URL and click go 2. Submit the score details | Given Details | Application should show a result page the shows the output "You have chance" and all the user interface components | Working as expected | Pass | The UI looks fine | Akash Stephen C |
| Result_page_02 | Functional | Result Page | Verify the UI elements | | 1. Enter URL and click go 2. Submit the score details | Given Details | Application should show a result page the shows the output "You don't have chance" and all the user interface | Working as expected | Pass | The UI looks fine and Interactive | Jarvin J |

8.2 User Acceptance Testing

8.2.1. Purpose of the 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).

8.2.2.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 | 5 | 2 | 1 | 2 | 10 |
| Duplicate | 0 | 1 | 2 | 1 | 4 |
| External | 4 | 2 | 2 | 1 | 9 |
| Fixed | 4 | 3 | 2 | 15 | 24 |
| Not Reproduc ed | 0 | 0 | 0 | 1 | 1 |
| Skipped | 0 | 1 | 0 | 1 | 2 |
| Won't Fix | 0 | 2 | 1 | 1 | 4 |
| Totals | 13 | 11 | 8 | 22 | 54 |

8.2.3.Test Case Analysis

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

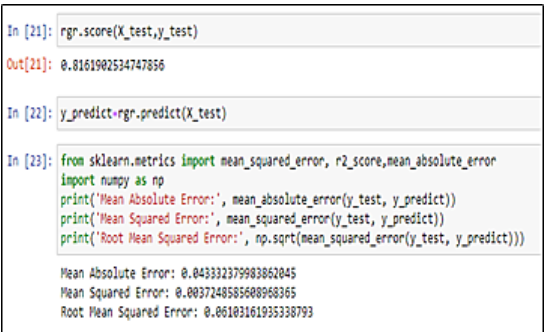
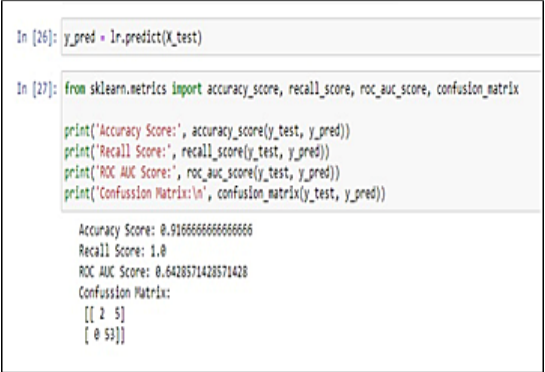
| Section | Total Cases | Not Tested | Fail | Pass |
|---------------------|-------------|------------|------|------|
| Print Engine | 5 | 0 | 1 | 4 |
| Client Application | 32 | 0 | 6 | 26 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 4 | 1 | 0 | 3 |
| Exception Reporting | 8 | 1 | 1 | 7 |
| Final Report Output | 5 | 1 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

CHAPTER 9

RESULTS

9.1 Performance Metrics

Model Performance Testing:

| S.No. | Parameter | Values | Screenshot |
|-------|-----------|---|--|
| 1. | Metrics | <p>Regression Model:</p> <p>MAE -0.043332</p> <p>MSE -0.0037248</p> <p>RMSE - 0.051031</p> <p>R2 score – 0.721402</p> <p>CLASSIFICATION MODEL</p> <p>Confusion matrix - $\begin{bmatrix} 2 & 5 \\ 0 & 53 \end{bmatrix}$</p> <p>Accuracy Score- 0.91666</p> <p>Classification Report-</p> <p>Recall Score-1.0</p> <p>ROC AUC Score-0.648571</p> |  <pre> In [21]: rgr.score(X_test,y_test) Out[21]: 0.8161902534747856 In [22]: y_predict=rgr.predict(X_test) In [23]: 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))) Mean Absolute Error: 0.043332379903862045 Mean Squared Error: 0.0037248585608968365 Root Mean Squared Error: 0.06103101935338793 </pre>  <pre> In [26]: y_pred = lr.predict(X_test) In [27]: 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)) Accuracy Score: 0.9166666666666666 Recall Score: 1.0 ROC AUC Score: 0.6428571428571428 Confusion Matrix: [[2 5] [0 53]] </pre> |

| | | | |
|----|----------------|--|--|
| 2. | Tune the Model | Hyperparameter Tuning- Validation Method- | <pre> In [64]: scores = cross_val_score(model, X_train, y_train, scoring='r2', cv=5) scores Out[64]: array([0.81813967, 0.77169539, 0.83989563, 0.74719974, 0.78589678]) In [65]: avg_score=scores.mean() In [67]: print ("Cross Validation Scores : ",scores) print ("Average CV Score : ",avg_score) print ("Number of CV Scores used in Average : ",len(scores)) Cross Validation Scores : [0.81813967 0.77169539 0.83989563 0.74719974 0.78589678] Average CV Score : 0.7925654408790849 Number of CV Scores used in Average : 5 </pre> |
|----|----------------|--|--|

CHAPTER 10

ADVANTAGES & DISADVANTAGES

10.1.Advantages

- It helps students in making decisions for choosing the right college.
- Here the chance for error occurrence is meager.
- It is fast, reliable, and efficient.
- It is very User-Friendly.
- Easy accessibility of data.
- Avoids data redundancy and inconsistency.

10.2.Disadvantages

- Required active internet connection.
- System will provide inaccurate results if data is entered incorrectly.

CHAPTER 11

CONCLUSION

In this project, we have developed a model to get the details from the students who want to choose a college for their higher studies. The students should enter details such as GRE score, TOFEL score, University Rating, SOP, LOR, CGPA, and Research details. As we have already trained the model with the dataset containing the previous year's admission details, this model predicts whether the student has the possibility to get admitted to college or not. If the students' scores satisfy the admission criteria, the result is shown as he/she has the chance to get admitted to the selected university. Else the result is shown as he/she doesn't have the chance to get admitted to the selected university.

CHAPTER 12

FUTURE SCOPE

- This project can be further enhanced by training the model with the data set of admission details from various other universities too.
- The User Interface can also be enhanced by including more graphical components
- Unique user profiles can be registered and given a login ID.
- Details of the universities within the country can also be used for training.

CHAPTER 13

APPENDIX

SOURCE CODE LINK:

https://drive.google.com/drive/folders/1DA1ls1o_XvVg4ry9cnxl-dNbXyHoL0UO?usp=share_link

GitHub link :

<https://github.com/IBM-EPBL/IBM-Project-30735-1660181843>

PROJECT DEMO VIDEO LINK :

YouTube link:

https://youtu.be/jhilpztf_5g

Google Drive link:

https://drive.google.com/file/d/1GAXnzw3Z4-poQZkWZy8Plr_nbeLG-zOL/view?usp=share_link