

# **UNIVERSITY ADMIT ELIGIBILITY PREDICTOR**

**(TEAM ID: PNT2022TMID32941)**

## **A PROJECT REPORT**

**Submitted by**

**SARANYA R (820419104061)**

**SANTHOSHINI M (820419104059)**

**TAMIL PRABHA R (820419104077)**

**VAISHNAVI P (820419104078)**

**in partial fulfillment for the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**ANJALAI AMMAL MAHALINGAM ENGINEERING COLLEGE  
KOVILVENNI-614 403**

**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
- 5.3 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

## **8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

## **9. RESULTS**

9.1 Performance Metrics

## **10. ADVANTAGES & DISADVANTAGES**

## **11. CONCLUSION**

## **12. FUTURE SCOPE**

## **13. APPENDIX**

Source Code

GitHub & Project Demo Link

# 1.INTRODUCTION

## 1.1 Project Overview :

Statistically, we have seen a lot of students pursue their education away from their native countries. Generally as the students don't have much of an idea about the procedures, requirements and details of the universities, they seek help from the education consultancy firms to help them successfully secure the admission in the universities which are best suitable for their profiles. For this, they have to invest huge amount of money as consultancy fees. The aim of this research is to develop a system using machine learning algorithm. UNIVERSITY PREDICTION would be the easiest mode to predict the university/colleges person is applicable for as well as it would unbiased and totally transparent. Individually would no more need to depend upon the consultancies who may be slightly deviated towards the list of colleges/university that may be having contract with them. Moreover applying to only that colleges/university where the student has genuine chance would even reduce application process. Additionally living expense of the area where colleges/university is located would also be provided on website

## 1.2 PURPOSE:

The people who will benefit the most from using this system are Indian students. Especially students looking to pursue their higher education from foreign universities, particularly in the United States. The administrator shall be able to access all the data stored in the application. Locations .The system will be available to all users from any location as long as they have an Internet connection. The administrator can also access the website from any location as long as he has the correct login credentials and access to the Internet. Provide customers access to the prediction model .Provide answers to most common FAQs regarding PG Admissions abroad Provide administrator access to all records Provide analysis of how the various academic factors affect university admission Other desired features of the system. Maintaining a profile for each user .Password protection for each account .This system is needed so as to answer the queries of students in a compete and concise manner as well as to provide them an as accurate as possible analysis of their chances of admissions to their dream universities.

## 2. LITERATURE SURVEY

### 2.1 Existing problem:

When the user wants to know whether he/she is eligible for getting admission in the universities. They need to visit the universities personally and get their previous cut off records. Then the user needs to analyze those records to know their chance of admit. It takes a lot of time and energy and the prediction will also be inaccurate. The proposed system will be able to overcome these shortcomings and gives the chance of admit for the user.

### 2.2 References:

1. C. Haythorhwaithe, M. de Laat, and S. Dawson, Introduction to the special issue on the learning analytics. American Behavioral Science,57(10):1371-1379,2013.
2. Liu Jinpeng. Research on the application of Data Mining Technology in Analysis of Examinee Wish, Henan University,2009.
3. Alpaydin,E. Introduction to Machine Learning,3rded;MIT press:Cambridge,MIT, USA,2010.
4. Kuncheva, LL combining pattern classifiers: Methods and Algorithms, 2nd ed; McGraw hill;John wiley&sons,Inc:Hoboken,NJ,USA,2014.
5. D.M Blei, A.Y. Ng, and M. I. Jordan, Latent Dirichlet allocation,Journal of Machine Learning Research,3:993-1022, 2003.
6. L. Breiman, Accuracy Predictors, Machine Learning, 24(2):123-140,1996.
7. Data Cleaning and Analytics, Machine Learning
8. Data Visualizaton, Machine Learning
9. Jupyter Notebook, Implementing the Algorithms, Machine Learning, <https://jupyter-notebook.readthedocs.io/en/stable/>

## Introduction:

This section includes the literature review of previous research on the assessment of student enrolment opportunities in universities. Numerous programs and studies have been carried out on topics relating to university admission used many machine learning models which helps the students in the admission process to their desired universities. Previous research done in this area used Naïve Bayes algorithm which will evaluate the success probability of student application into a respective university but the main drawback is they didn't consider all the factors which will contribute in the student admission process like TOEFL/IELTS, SOP, LOR and under graduate score(Chithra Apoorva D.A march 2020).

## Problem Understanding

Initially first we have to spend some time on what are the problems or concerns students having during their pre admission period and we should set the solutions to those problems as objectives of this research.

**Data Understanding:** Data should be collected from multiple sources like yocket and also consider all the factors including which will play a tiny role in student admission process.

**Evaluation:** Developed models are evaluated according to their accuracy scores. Once the model is finalised that model will be merged with node red for final deployment.(Malepati chanduNath, march2020)

## Data collection

The way toward get together information relies upon sort of undertaking, for a ML project, real time information is utilized. The information index can be gathered from differences sources like a document, data set, sensor and different sources and some free informational collection from web can be utilized.

## **Pre-Processing**

Information pre-processing is a cycle of cleaning the raw information i. e the information is gathered in reality and is changed over to a perfect dataset. There are certain steps executed to change Over the data into a little clean data collection and make it practical for examination, this piece of the Interaction is called as information pre-processing (Krithika cs October 2021).

The greater part of this present reality information is chaotic, as :

- Missing Data
- Noisy Data
- Inconsistent Data
- Conversion of Data
- Ignoring the missing Qualities
- Filling the missing Qualities
- Detection of Exceptions
- Feature Extraction

## **Data virtualization**

Data Virtualization is the representation of data in a graph, chart, or other visual format. Data virtualization by using the histogram and analyzing the data by the graph. The top three featues that affect the chance of Admit are:

- CGPA
- GRE Score
- TOEFL Score

## 2.3 Problem Statement Definition:

### Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

|                            |   |   |
|----------------------------|---|---|
| <b>I am</b>                | <small>Describe customer with 3-4 key characteristics - who are they?</small>                                 | Describe the customer and their attributes here                             |
| <b>I'm trying to</b>       | <small>List their outcome or "job" the care about - what are they trying to achieve?</small>                  | List the thing they are trying to achieve here                              |
| <b>but</b>                 | <small>Describe what problems or barriers stand in the way - what bothers them most?</small>                  | Describe the problems or barriers that get in the way here                  |
| <b>because</b>             | <small>Enter the "root cause" of why the problem or barrier exists - what needs to be solved?</small>         | Describe the reason the problems or barriers exist                          |
| <b>which makes me feel</b> | <small>Describe the emotions from the customer's point of view - how does it impact them emotionally?</small> | Describe the emotions the result from experiencing the problems or barriers |



| Problem Statement (PS) | I am (Customer)                | I'm trying to            | But                    | Because                 | Which makes me feel |
|------------------------|--------------------------------|--------------------------|------------------------|-------------------------|---------------------|
| PS-1                   | 12 <sup>th</sup> grade student | get admission details in | There is no updates in | of no real time updates | stressed            |



|      |               |   |  |                          |           |
|------|---------------|---|--|--------------------------|-----------|
|      |               | website   | seat availability                      |                          |           |
| PS-2 | a UG graduate | Check my eligibility of university for higher studies | There is no proper guidance in website | of no proper information | difficult |

## 3. IDEATION & PROPOSED SOLUTION

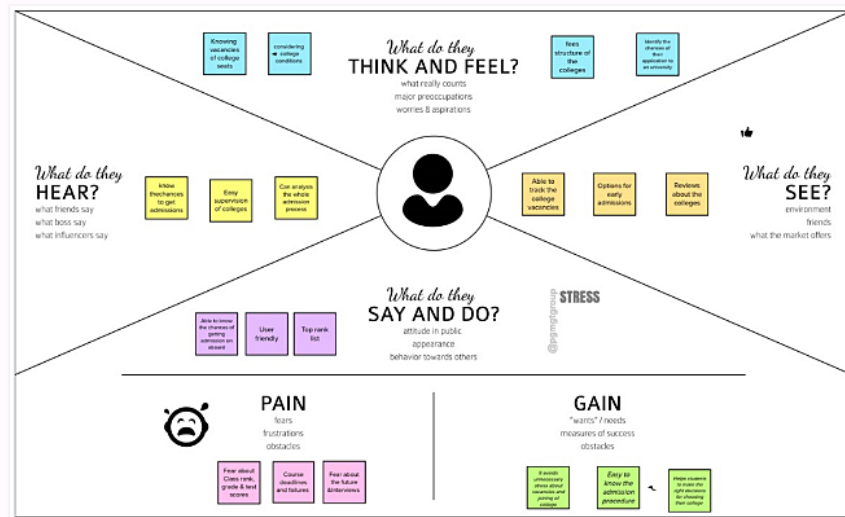
### 3.1 EMPATHY MAP CANVAS:

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 helps 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.

Empathy Map:

### UNIVERSITY ADMIT ELIGIBILITY PREDICTOR



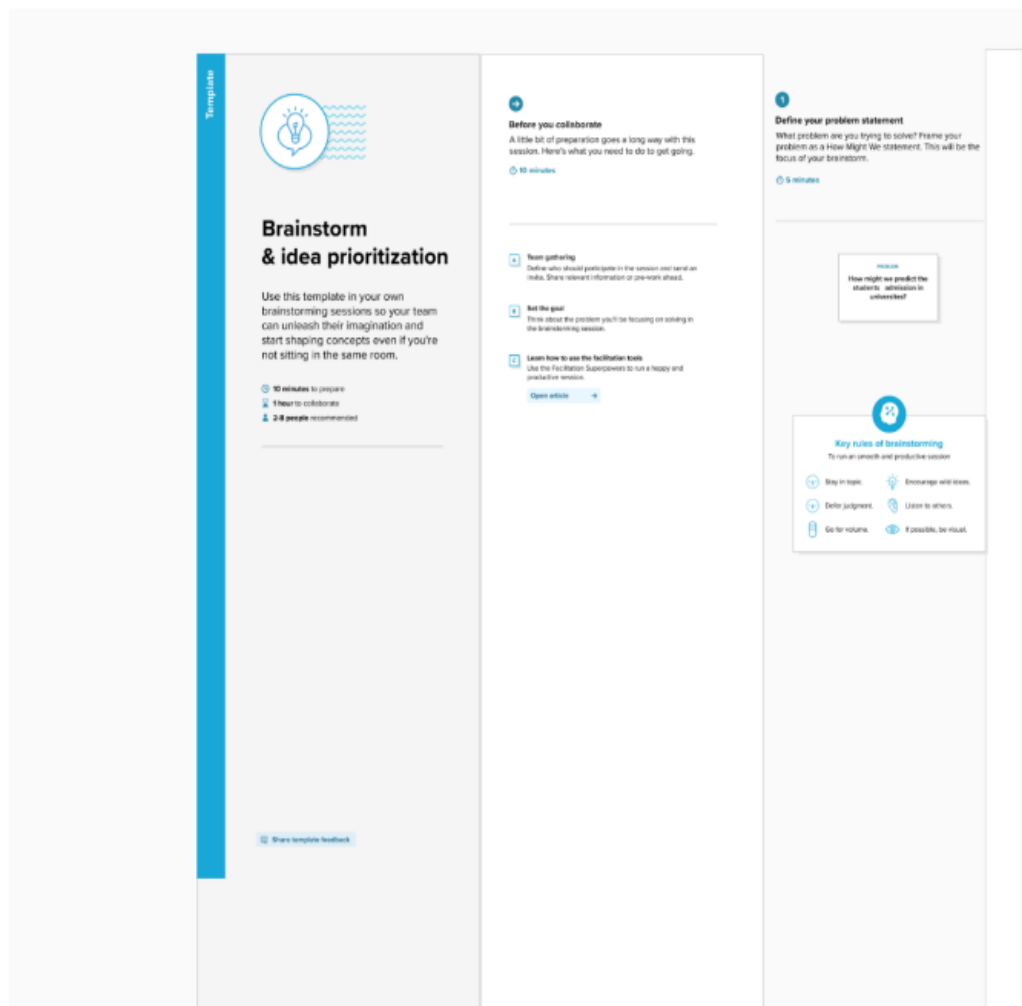
## 3.2 IDEATION & BRAINSTORMING:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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

Reference: <https://www.mural.co/templates/empathy-map-canvas>

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



## Step-2: Brainstorm, Idea Listing and Grouping.

### 3

#### Group ideas

Now some making your ideas into clusters by similar or related notes on you got. In the last 10 minutes, give each cluster a sentence that says: If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

**Service**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Self-Service**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Network**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Self-Service**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Based on records**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Based on marks**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Based on fees structure**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Based on university distance**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Based on special talents**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**Based on collecting information**

- 1. Create a new service
- 2. Create a new service
- 3. Create a new service
- 4. Create a new service
- 5. Create a new service
- 6. Create a new service
- 7. Create a new service
- 8. Create a new service
- 9. Create a new service
- 10. Create a new service

**TIP**

Don't forget to check the service you are creating is not already existing in the market. If it is, then you need to think of a way to make it better than the existing one.

## Step-3: Idea Prioritization.

### 4

#### Prioritize

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

20 minutes

**Importance**

How much of these ideas could you get done without any difficulty or cost, which would have the most positive impact?

**Feasibility**

Regardless of their importance, which ideas are more feasible than others? (Cost, time, effort, complexity, etc.)

**TIP**

Participants can use their laptops to enter all ideas. Sticky notes should go on the grid. The facilitator can confirm the grid by using the laser pointer (holding the key) on the keyboard.

### 3.3 PROPOSED SOLUTION:

| S.No. | Parameter                                | Description  |
|-------|--|--|
| 1.    | Problem Statement (Problem to be solved) | predicts a user's chances of admissions in the university of their choice for Studies .  |
| 2.    | Idea / Solution description              | Addressed machine learning models to predict the chance of a student to be admitted to a programs. This will assist students to know in advance if they have a chance to get accepted.   |
| 3.    | Novelty / Uniqueness                     | Improve the operational efficiency by improving the quality of the process.<br>One of the important objectives of the admission system is communicate with all the students scattered geographically.  |
| 4.    | Social Impact / Customer Satisfaction    | Student satisfaction at university is receiving increasing attention. While academic discipline has been associated with student satisfaction in many studies.   |
| 5.    | Business Model (Revenue Model)           | A machine leaning based system built on a linear regression model using the data set available on kaggle for predicting chances of admissions for indian students hoping to pursue their post graduate studies abroad  |
| 6.    | Scalability of the Solution              | UNIPREDICT are here to provide a solution to that problem. Not only do we provide a single platform that documents all the requirements as well as the different tiers of universities, but our website also incorporates an AI Model that was built after considering many leading Machine Learning Algorithms, to provide the most accurate prediction of how much of a chance of admissions does a student's current grades and other academic transcripts allow them in the tier of universities of their choice |

## 3.4 PROBLEM SOLUTION FIT:

Project Title: University Admit Eligibility Predictor

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID32941

|                            |  |  |  |                           |
|----------------------------|--|--|--|---------------------------|
| Define CS, fit into CC     | <b>1. CUSTOMER SEGMENT(s):</b> <span>CS</span><br><br>Hsc finished students or UG completed students   | <b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span><br><br>Network connection ,Cost ,Time ,PoorKnowledge, Lack of Resoures.   | <b>5. AVAILABLE SOLUTION:</b> <span>AS</span><br><br>Seat allotment,Eligibility criteria, Previous year cut off , Exam scores likeGRE,TOEFL,GATE etc.  | Explore AS, differentiate |
|                            | <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? There could be more than one, explore different sides.<br><br>High Fees structure , placement Opportunites & Training ,Courses Offered , Advanced Technology , Career development programmes | <b>9. PROBLEM ROOT CAUSE</b> <span>RC</span><br>What is the real reason that this problem exists? What is the back story behind the need to do this job?<br><br>Lack of Placement Opportunites and Not meeting the Expected cut off. Due to high Competition & less opportunities results in the difficulty to choose a right college  | <b>7. BEHAVIOUR</b> <span>BE</span><br>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)<br><br>Referring Articles & checking websites ,Visiting College premises , Academic Performes , Fees structure and enquiring Alumni.   |                           |
| Focus on J&P, tap into BE. | <b>3. TRIGGERS</b> <span>TR</span><br>Challenging high school curriculum.Meaningfu involvement in extracurricular activities.<br>Interviews that support the students strength and good character.<br><br>To put a road map for future career &to get high paying job offers.  | <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>This idea help the students to get the list of colleges by comparing the studendsmarks&colleges cut off & predicting admission probability. Here the chance of occurrence of error is less when compared to existing system. It is fast, efficient & reliable. It helps you to understand as to how your profile can | <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>Referring articles & websites , Advertisements, College Reputation, Course Available, List of Top College,Eligibility Criteria, Previous Placements.<br><br><b>8.2OFFLINE</b><br>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.<br><br>Visiting College campuses, Food & Accommodation, Transport & lab facilities, |                           |
|                            | <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>Confused, Anxious about peer group, Exciting & Enthusiastic & Friendly Faculties           | be furture improved to secure an admit in your target college.   | Enquiring College students, Speaking to Academic representatives.  |                           |

## 4. REQUIREMENT ANALYSIS:

### 4.1 Functional Requirements:

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task)   |
|--------|-------------------------------|--|
| FR-1   | User Registration             | Registration through Form.<br>Registration through Gmail.<br>Registration through mobile number.   |
| FR-2   | User Confirmation             | Confirmation via Gmail.<br>Confirmation via Password.  |
| FR-3   | User Authentication           | Authentication through username.<br>Authentication through Password.<br>Authentication through OTP.  |
| FR-4   | User recovery                 | Recover through forget password.<br>Recover through security question.   |
| FR-5   | User facility                 | User can enter the personal details.<br>User can enter the academic details.<br>User can give ratings/review.<br>User can update their information.                      |
| FR-6   | Special features              | Only admin can have all the access.<br>Recommend the university to the user private.<br>Admin can reply user queries.<br>Keep their student details very confidentially. |

### Non-functional Requirements:

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

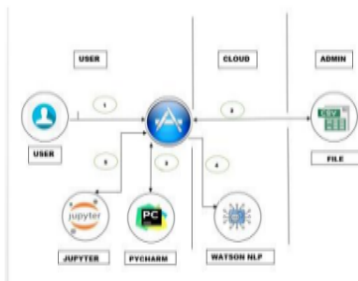
| FR No. | Non-Functional Requirement | Description   |
|--------|----------------------------|---|
| NFR-1  | Usability                  | Friendly user interface.<br>Higher performance.<br>Fast prediction.                   |
| NFR-2  | Security                   | User can use password for security.<br>Both admin and student use password to protect |

|       |                     |   |
|-------|---------------------|---|
|       |                     | their information very confidential.  |
| NFR-3 | <b>Reliability</b>  | Website response quickly.   |
| NFR-4 | <b>Performance</b>  | Perform technically stable and secure.<br>More than two number of users support at a time.<br>User cannot visit any unwanted ads and blog.<br>student should easy to navigate.    |
| NFR-5 | <b>Availability</b> | Available in all operating system like Windows, Mac, Linux, IOS etc and also available in android so student can use both mobile phones and system.<br>Available in all the time. |
| NFR-6 | <b>Scalability</b>  | Website will be in dynamic nature so user can update the details anytime.   |

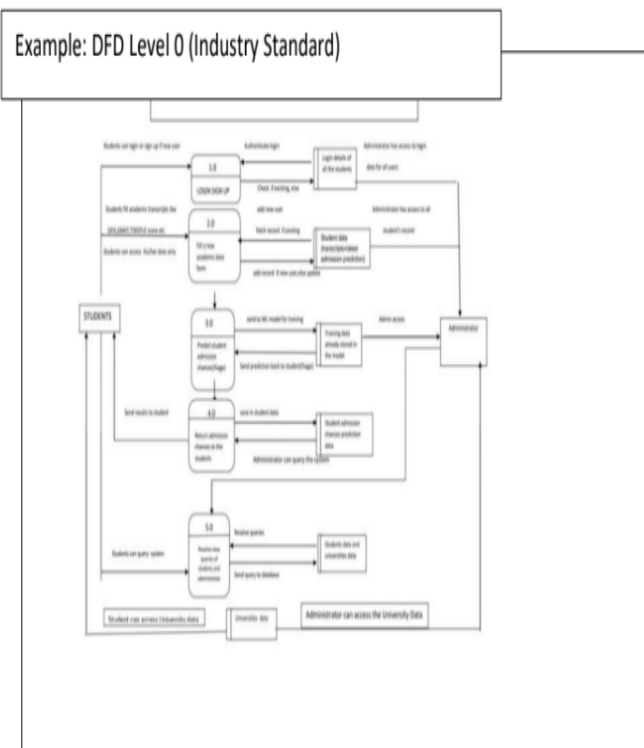
## 5. PROJECT DESIGN:

### 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.



1. Natural Language Understanding Service and Starts the app.
2. User Selects Data(csv) file to process and load.
3. Jupyter Lab Extracts text from the data file and build, render the app.
4. Extracted text is passed to Watson NLU for enrichment
5. Enriched data is visualized in the UI using the Pycharm User configures credentials for the Watson



## 5.2 Solution & Technical Architecture:

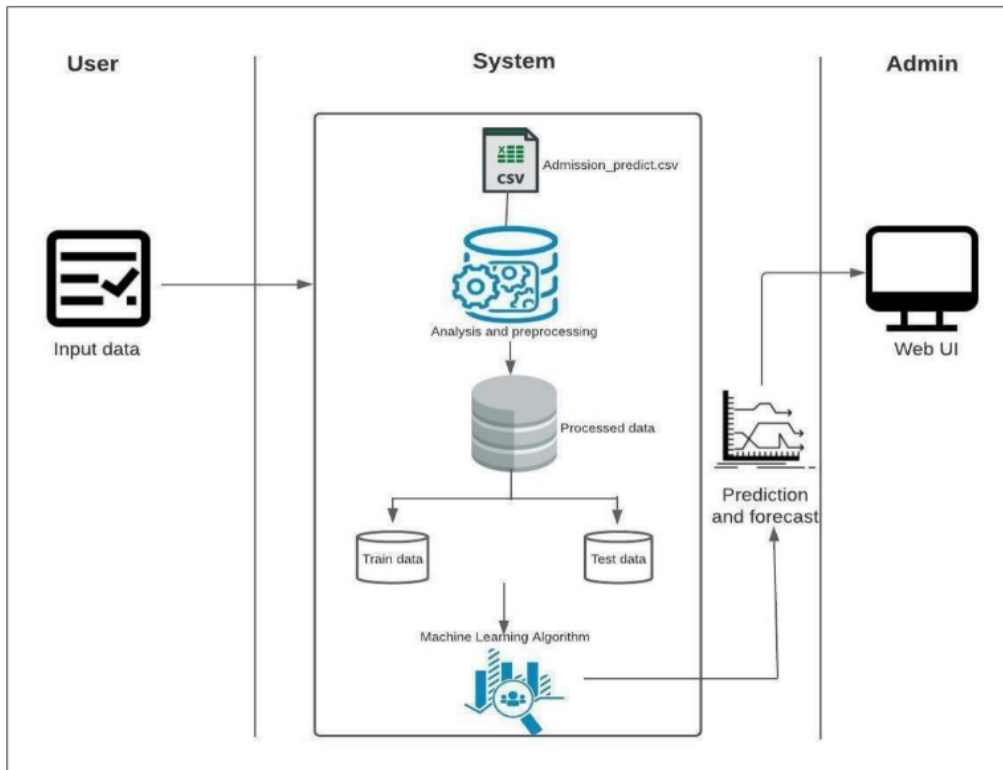




Table-1: Components and Technologies:

| S.No | Component               | Description  | Technology               |
|------|-------------------------|--|--------------------------|
| 1.   | User Interface          | The user interacts with the application through a Web UI   | HTML, CSS, Python, Flask |
| 2.   | Application Logic-1     | Logic for collecting the input from the user   | Python                   |
| 3.   | Application Logic-2     | Integrating Machine Learning model with our application  | Python                   |
| 4.   | Database                | Numeric data   | MySQL                    |
| 5.   | File Storage            | To store files such as prediction report   | Local Filesystem         |
| 10.  | Machine Learning Model  | Predictive modelling is a mathematical process used to predict future events or outcomes by analysing patterns in a given set of input data. | Predictive Modelling     |
| 11.  | Infrastructure (Server) | Application Deployment on Local System<br>Local Server Configuration: Built-in Flask web server  | Flask, Web server        |

Table-2: Application Characteristics:

| S.No | Characteristics         | Description   | Technology                      |
|------|-------------------------|---|---------------------------------|
| 1    | Open-Source Framework   | Flask   | Micro web framework with python |
| 2.   | Security implementation | Http authentication, Session based authentication   | Flask security                  |
| 3.   | Scalable                | Size is everything, and Flask's status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. Its simplicity of use and few dependencies enable it to run smoothly even as it scales up and up. | Flask                           |
| 4.   | Availability            | Higher compatibility with latest technologies and allows customization  | Flask                           |
| 5.   | Performance             | Integrated support for unit testing.<br><ul style="list-style-type: none"> <li>• RESTful request dispatching.</li> <li>• Uses Jinja templating.</li> <li>• Support for secure cookies</li> </ul>  | Flask                           |

## 5.3 User Stories:

### User Stories

| User Type                             | Functional Requirement (Epic) | User Story Number | User Story / Task   | Acceptance criteria  | Priority | Release  |
|---------------------------------------|-------------------------------|-------------------|---|--|----------|----------|
| Customer (Mobile user/Desktop/Laptop) | 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                              | Low      | Sprint-2 |
|                                       |                               | USN-4             | As a user, I can register for the application through Gmail   | I can register & access the dashboard                              | Medium   | Sprint-1 |
|                                       | Login/Sign Up                 | USN-5             | As a user, I can log into the application by entering email & password                                    | I can register & access the dashboard                              | High     | Sprint-1 |
|                                       | Dashboard                     | USN-6             | As a User, I can view the essential details about System like user manual, Settings, about.               | I can access the dashboard.  |          | Sprint-1 |
| Customer (Web user)                   | Login/Sign Up                 | USN-7             | As a User, I can use the currently updated version of the web application.                                | I can access through the website.                                  | High     | Sprint-1 |
| Customer Care Executive               | Chat Box/ChatBot              | USN-8             | As a Customer care Executive clarify the FAQ's and User Questions/Doubts About the system.                | I can access my Chatbox /Chat Bot                                  | High     | Sprint-2 |
| Administrator                         | Universities Data             | USN-9             | As a Customer care Executive clarify the FAQ's and User Questions/Doubts About the system.                | I can receive the confirmation message from student's credentials. | High     | Sprint-3 |
|                                       |                               | USN-10            | As an Admin, I can active and de-active student's status.   | I can access my account /Login Credentials                         | Medium   | Sprint-3 |
|                                       |                               | USN-11            | As an Admin, I can Block or Unblock the User.   | I can access my account /Login Credentials                         | High     | Sprint-3 |
|                                       |                               | USN-12            | As an Admin, I can Edit the user data.  | I can Modify the data via University data.                         | Medium   | Sprint-3 |

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation:

Use the below template to create product backlog and sprint schedule

| Sprint    | Functional Requirement (Epic) | User Story Number | User Story / Task   | Story Points | Priority | Team Members  |
|-----------|-------------------------------|-------------------|---|--------------|----------|---|
| Sprint-1  | Registration                  | USN-1             | As a user, I can register for the application by entering my email, password, and confirming my password. | 2            | High     | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint-1  | Login                         | USN-2             | As a user, I will receive confirmation email once I have registered for the application                   | 1            | High     | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint-1  | Data collection               | USN-3             | Gathering the information from various resources  | 1            | Medium   | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint-1  | Data pre-processing           | USN-4             | To convert and clean the raw data   | 1            | High     | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint-2  | Model building                | USN-5             | Using cleaned dataset ,Model can be build by using Machine learning algorithm                             | 3            | High     | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint -2 |                               | USN-6             | Train the classification model.   | 5            | High     | R. Saranya<br>M.Santhoshini                                 |

| Sprint    | Functional Requirement (Epic)                             | User Story Number | User Story / Task                                  | Story Points | Priority | Team Members  |
|-----------|---|-------------------|--|--------------|----------|---|
|           |   |                   |  |              |          | R.Tamilprabha<br>P.Vaishnavi                                |
| Sprint -3 | Application Building                                      | USN-7             | Build the python code and run the application      | 5            | Medium   | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint -3 |   | USN-8             | Predicted results has sent to user registered mail | 2            | Low      | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint -4 | Implementation of the application and deployment on cloud | USN-9             | Deployed on IBM Cloud                              | 3            | High     | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |

## 6.2 Sprint Delivery Schedule:

| Sprint    | Functional Requirement (Epic)                             | User Story Number | User Story / Task                                  | Story Points | Priority | Team Members  |
|-----------|---|-------------------|--|--------------|----------|---|
|           |   |                   |  |              |          | R.Tamilprabha<br>P.Vaishnavi                                |
| Sprint -3 | Application Building                                      | USN-7             | Build the python code and run the application      | 5            | Medium   | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint -3 |   | USN-8             | Predicted results has sent to user registered mail | 2            | Low      | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |
| Sprint -4 | Implementation of the application and deployment on cloud | USN-9             | Deployed on IBM Cloud                              | 3            | High     | R. Saranya<br>M.Santhoshini<br>R.Tamilprabha<br>P.Vaishnavi |

## 6.3 Reports from JIRA:

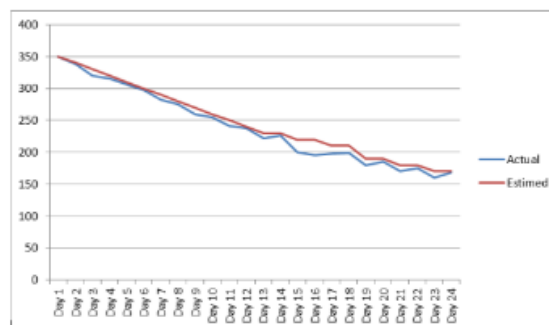
Velocity:

$$AV = \frac{\text{Sprint duration}}{\text{Velocity}} = \frac{20}{10}$$

$$AV = 20/6 = 3.33$$

### BURNDOWN CHART:

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, burndown charts can be applied to any project containing measurable progress over time.



## 7. CODING & SOLUTIONING (Explain the features added in the project along with code)

### 7.1 Feature 1:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeRegressor
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline

import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your
# credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
                              ibm_api_key_id='kcBR14_NqTmdFtQlfKj8ZyBvaM6rDh_OK5RuQ8tng1rh',
                              ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
```

```
config=Config(signature_version='oauth'),
endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')
```

```
bucket = 'newproject-donotdelete-pr-hbgeifgektjysh'
object_key = 'Admission_Predict.csv'
```

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

```
data = pd.read_csv(body)
data.head()

data.head()
```

Out[72]:

|   | GRE Score | TOEFL Score | University Rating | SOP | LOR | CGPA | Research | Chance of Admit |
|---|-----------|-------------|-------------------|-----|-----|------|----------|-----------------|
| 0 | 337       | 118         | 4                 | 4.5 | 4.5 | 9.65 | 1        | 0.92            |
| 1 | 324       | 107         | 4                 | 4.0 | 4.5 | 8.87 | 1        | 0.76            |
| 2 | 316       | 104         | 3                 | 3.0 | 3.5 | 8.00 | 1        | 0.72            |
| 3 | 322       | 110         | 3                 | 3.5 | 2.5 | 8.67 | 1        | 0.80            |
| 4 | 314       | 103         | 2                 | 2.0 | 3.0 | 8.21 | 0        | 0.65            |

```
data.shape
```

```
(400, 8)
```

```
data.info()
```

```
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   GRE Score              400 non-null   int64   
1   TOEFL Score            400 non-null   int64   
2   University Rating      400 non-null   int64   
3   SOP                    400 non-null   float64  
4   LOR                    400 non-null   float64  
5   CGPA                   400 non-null   float64  
6   Research               400 non-null   int64   
7   Chance of Admit        400 non-null   float64  
dtypes: float64(4), int64(4)
memory usage: 25.1 KB
```

|       | GRE Score  | TOEFL Score | University Rating | SOP        | LOR        | CGPA       | Research   | Chance of Admit |
|-------|------------|-------------|-------------------|------------|------------|------------|------------|-----------------|
| count | 400.000000 | 400.000000  | 400.000000        | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000      |
| mean  | 316.807500 | 107.410000  | 3.087500          | 3.400000   | 3.452500   | 8.598925   | 0.547500   | 0.724350        |
| std   | 11.473646  | 6.069514    | 1.143728          | 1.006869   | 0.898478   | 0.596317   | 0.498362   | 0.142609        |
| min   | 290.000000 | 92.000000   | 1.000000          | 1.000000   | 1.000000   | 6.800000   | 0.000000   | 0.340000        |
| 25%   | 308.000000 | 103.000000  | 2.000000          | 2.500000   | 3.000000   | 8.170000   | 0.000000   | 0.640000        |
| 50%   | 317.000000 | 107.000000  | 3.000000          | 3.500000   | 3.500000   | 8.610000   | 1.000000   | 0.730000        |
| 75%   | 325.000000 | 112.000000  | 4.000000          | 4.000000   | 4.000000   | 9.062500   | 1.000000   | 0.830000        |
| max   | 340.000000 | 120.000000  | 5.000000          | 5.000000   | 5.000000   | 9.920000   | 1.000000   | 0.970000        |

```
Null=data.isnull()
```

```
Null.sum()
```

```
data.duplicated().sum()
```

```
data.drop('Serial No.', axis='columns', inplace=True)
```

```
data.head()
```

```
sns.displot(x=data["GRE Score"], kde=True, color='darkblue')
```

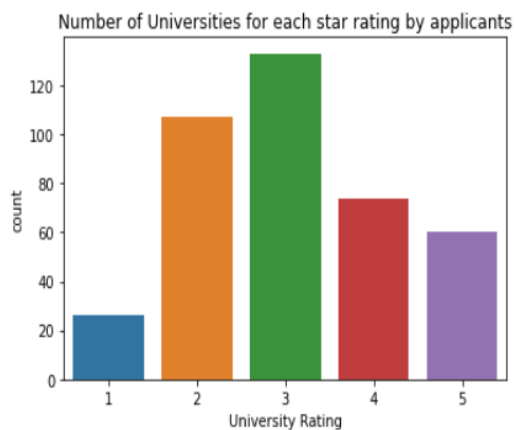
```
plt.title("GRE score distribution with density distribution");
```

```
sns.displot(x=data["TOEFL Score"], kde=True, color='olive')
```

```
plt.title("TOFEL score distribution with density distribution");
```

```
sns.countplot(x=data["University Rating"]);
```

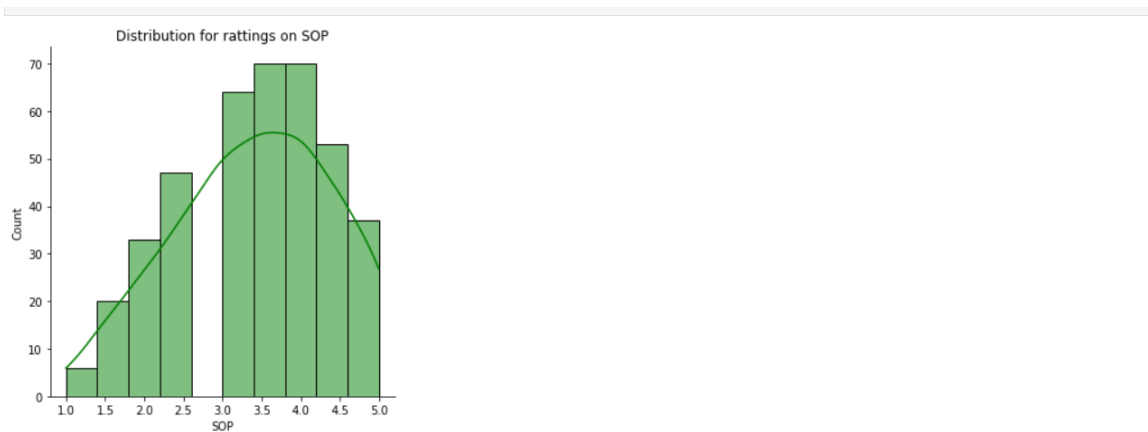
```
plt.title("Number of Universities for each star rating by applicants");
```



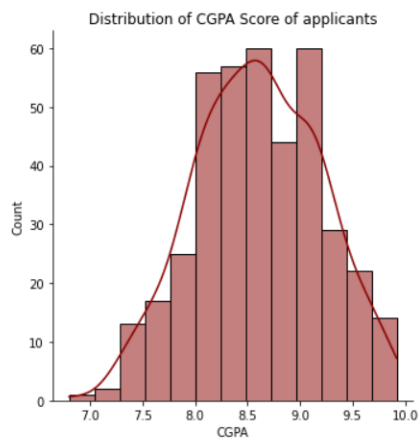
```
sns.displot(x=data["SOP"], kde=True, color='green');
```

```
plt.title("Distribution for rattings on SOP");
```

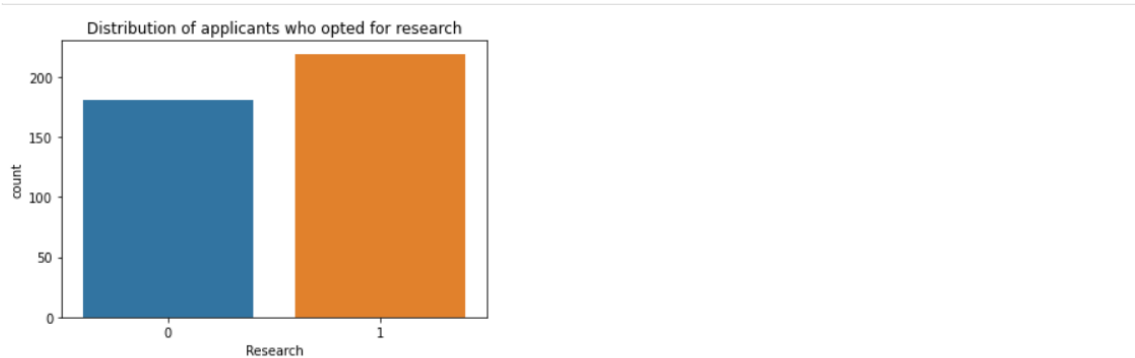
```
sns.displot(x=data["LOR "], kde=True, color='red');  
plt.title("Distribution of rattings given to Letter of recommendation of applicants");
```



```
sns.displot(x=data["CGPA"], kde=True, color='darkred');  
plt.title("Distribution of CGPA Score of applicants");
```

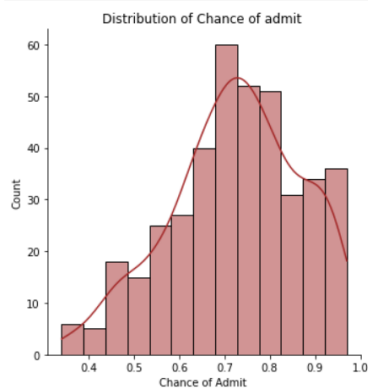


```
sns.countplot(x=data["Research"]);  
plt.title("Distribution of applicants who opted for research");
```

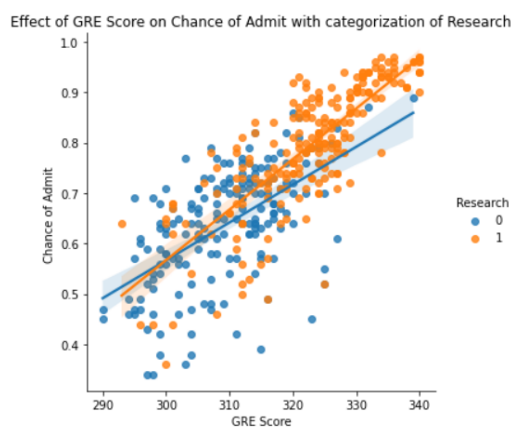




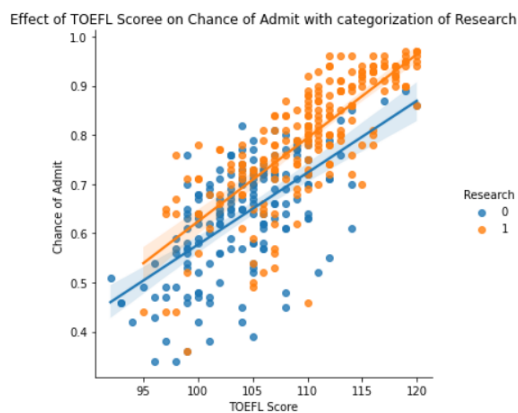
```
sns.displot(x=data["Chance of Admit "], kde=True, color='brown');
plt.title("Distribution of Chance of admit ");
```



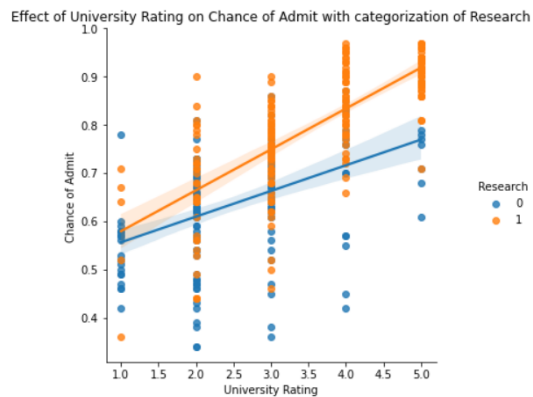
```
sns.lmplot(y="Chance of Admit ", x = "GRE Score", hue="Research", data=data);
plt.title("Effect of GRE Score on Chance of Admit with categorization of Research");
```



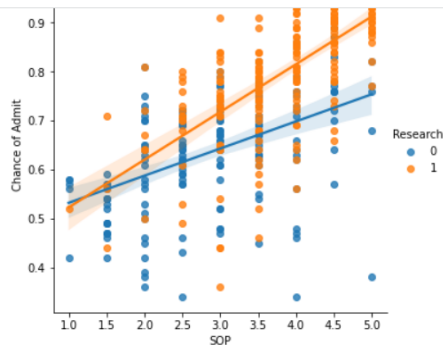
```
sns.lmplot(y="Chance of Admit ", x = "TOEFL Score", hue="Research", data=data);
plt.title("Effect of TOEFL Scoree on Chance of Admit with categorization of Research");
```



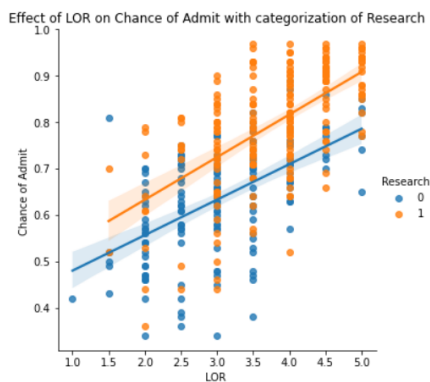
```
sns.lmplot(y="Chance of Admit ", x = "University Rating", hue="Research", data=data);
plt.title("Effect of University Rating on Chance of Admit with categorization of Research");
```



```
sns.lmplot(y="Chance of Admit ", x = "SOP", hue="Research", data=data);
plt.title("Effect of SOP on Chance of Admit with categorization of Research");
```

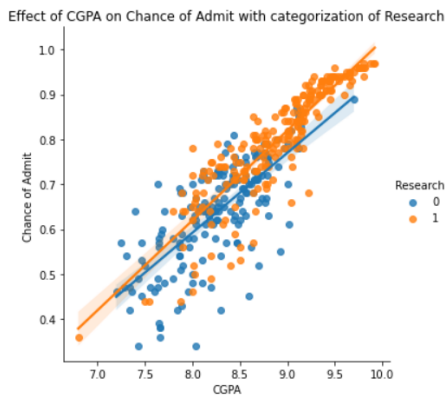


```
sns.lmplot(y="Chance of Admit ", x = "LOR ", hue="Research", data=data);
plt.title("Effect of LOR on Chance of Admit with categorization of Research");
```



```
sns.lmplot(y="Chance of Admit ", x = "CGPA", hue="Research",data=data);
```

```
plt.title("Effect of CGPA on Chance of Admit with categorization Research");
```



```
data = data.reset_index(drop=True)
```

```
X = data[data.columns.drop('Chance of Admit ')]
```

```
y = data["Chance of Admit "]
```

```
X.shape
```

```
(400, 7)
```

```
y.shape
```

```
(400,)
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=100)
```

```
X_train.shape
```

```
(320, 7)
```

```
X_test.shape
```

```
(80, 7)
```

```
#model
```

```
linear_regression = LinearRegression()
```

```
#training
```

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

```
LinearRegression()
```

```
model = LinearRegression(normalize=True)
model.fit(X_test, y_test)
# model.score(X_test, y_test)
print ("Accuracy : ",model.score(X_test, y_test)*100)
methodDict = {}
methodDict['Linear Regression'] = model.score(X_test, y_test)*100
Accuracy : 80.72164388568929
```

```
from sklearn.ensemble import RandomForestRegressor
rgr=RandomForestRegressor()
rgr.fit(X_train,y_train)
```

```
RandomForestRegressor()
```

```
print ("Accuracy : ",rgr.score(X_test, y_test)*100)
methodDict['Random Forest'] = rgr.score(X_test, y_test)*100
Accuracy : 73.21905063227847
```

## 7.2 Feature 2:

```
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 = "YXptRh7MuUVmneQk-4sC31DSfwlqjHGTxqjm0T3xdN-B"
```

```

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
    "apikey": API_KEY,
    "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
payload_scoring = {
    "input_data": [{"fields": ['GRE Score',
                              'TOEFL Score',
                              'University Rating',
                              'SOP',
                              'LOR ',
                              'CGPA',
                              'Research'],
                    "values": [arr]
                   }]
}

response_scoring = requests.post(
    'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/cf6a2ef7-ab21-4035-bb11-559230ec1d16/predictions?version=2022-11-18',
    json=payload_scoring,
    headers=header
).json()

result = response_scoring['predictions'][0]['values']

if result[0][0] > 0.6:
    return redirect(url_for('chance', percent=result[0][0] * 100))
else:
    return redirect(url_for('no_chance', percent=result[0][0] * 100))
else:
    return redirect(url_for("demo2"))

@app.route("/home")
def demo2():
    return render_template("demo2.html")

@app.route("/chance/<percent>")

```

```
def chance(percent):
    return render_template("chance.html", content=[percent])

@app.route("/no_chance/<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()
```

## 8.TESTING

### 8.1 Test Cases:

University Admission Eligibility Prediction System

**Enter your details and get probability of your admission**

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

**Enter the details**

GRE Score: 250 to 340

TOFEL Score: 50 to 120

University Rating: 1 to 5

SOP: 1 to 5

LOR: 1 to 5


CGPA: 5 to 10

Research: ☐ Yes ☒ No

**Predict**

deploy full link - rumprabha200 x University Admit Eligibility Predic x +

127.0.0.1:5000/nochance/48.679234349752875



You have a LOW / NO chance  
The model has predicted that you only have  
**48.679234349752875%** chance

Go Back


Type here to search

21:35  
19-11-2022

deploy full link - rumprabha200 x University Admit Eligibility Predic x +

127.0.0.1:5000/chance/81.64360350248715

University Admission Eligibility Prediction System



You Have Chance  
The model has predicted that you have  
**81.64360350248715%** chance

Go Back

Type here to search

21:36  
19-11-2022

## 9. RESULTS

### 9.1 Performance Metrics:

Model Performance Testing

Project team shall fill the following information in model performance testing template.

| S.No. | Parameter | Values  | Screenshot  |  |        |                   |   |     |          |   |     |          |   |      |          |   |          |          |
|-------|-----------|---|---|--|--------|-------------------|---|-----|----------|---|-----|----------|---|------|----------|---|----------|----------|
| 1.    | Metrics   | <p><b>Regression Model:</b><br/>MAE -0.043051, MSE - 0.003313, RMSE - 0.057560, R2 score - 0.807216</p> <p><b>Classification Model:</b><br/>Confusion Matrix ,<br/>Accuracy Score- 0.872 &amp;<br/>Classification Report.</p> | <pre>In [47]: mae = metrics.mean_absolute_error(y_test, predlinear) mse = metrics.mean_squared_error(y_test, predlinear) rmse = np.sqrt(mse) # or mse**(0.5) r2 = metrics.r2_score(y_test, predlinear)</pre> <pre>In [48]: chart = { 'Metric':["MAE", "MSE", "RMSE", "R2-SCORE"], 'LINEAR_REGRESSION':[mae,mse,rmse,r2], } chart = pd.DataFrame(chart)</pre> <pre>In [49]: display(chart)</pre> <table><thead><tr><th></th><th>Metric</th><th>LINEAR_REGRESSION</th></tr></thead><tbody><tr><td>0</td><td>MAE</td><td>0.043051</td></tr><tr><td>1</td><td>MSE</td><td>0.003313</td></tr><tr><td>2</td><td>RMSE</td><td>0.057560</td></tr><tr><td>3</td><td>R2-SCORE</td><td>0.807216</td></tr></tbody></table> <pre>In [43]: model = LinearRegression(normalize=True) model.fit(X_test, y_test) # model.score(X_test, y_test) predlinear = model.predict(X_test) print ("Accuracy : ",model.score(X_test, y_test)*100) methodDict = {} methodDict['Linear Regression'] = model.score(X_test, y_test)*100  Accuracy : 80.7216438856893</pre> |  | Metric | LINEAR_REGRESSION | 0 | MAE | 0.043051 | 1 | MSE | 0.003313 | 2 | RMSE | 0.057560 | 3 | R2-SCORE | 0.807216 |
|       | Metric    | LINEAR_REGRESSION   |   |  |        |                   |   |     |          |   |     |          |   |      |          |   |          |          |
| 0     | MAE       | 0.043051  |   |  |        |                   |   |     |          |   |     |          |   |      |          |   |          |          |
| 1     | MSE       | 0.003313  |   |  |        |                   |   |     |          |   |     |          |   |      |          |   |          |          |
| 2     | RMSE      | 0.057560  |   |  |        |                   |   |     |          |   |     |          |   |      |          |   |          |          |
| 3     | R2-SCORE  | 0.807216  |   |  |        |                   |   |     |          |   |     |          |   |      |          |   |          |          |



|    |                |   |  |
|----|----------------|---|--|
| 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> |
|----|----------------|---|--|

## 10. ADVANTAGES & DISADVANTAGES

### Advantages:

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

### Disadvantage:

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

## 11. CONCLUSION:

This system, being the first we have created in PHP, has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself. Debugging can sometimes be a relatively straight forward process, or rather finding out what you must debug can be. Since so many parts of the admission system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first. Language used must be simple and easy to understand and compatibility is paramount. If this system were not designed as an entirely web-based application, it would not have been possible to recreate its current state of portability. Overall, the system performs well, and while it does not include all of the features that may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

## 12. FUTURE SCOPE:

The future scope of this project is very broad. Few of them are:

- This can be implemented in less time for proper admit eligibility predictor process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.

## 13. APPENDIX:

GITHUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-45633-1660731387>

DEMO LINK:

<https://youtu.be/b8z5USc6OJc>