FINAL DELIVERABLES - PROJECT REPORT

UNIVERSITY ADMISSION ELIGIBILITY PREDICTOR

DOMAIN	APPLIED DATA SCIENCE
TOPIC	UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
TEAM ID	PNT2022TMID14806
TEAM MEMBERS	1. JAGADEESH KUMAR R - TEAM LEAD 2. SARAVANAN P 3. GOWTHAM M 4. OHMPRAKASH V
SUBMISIION DATE	19/11/2022

UNIVERSITY ADMISSION PREDICTION SYSTEM USING MACHINE LEARNING

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PROJECT OVERVIEW

The project uses a machine-learning model to estimate, using information like marks and other details, whether the user is qualified for admission to the rating universities that have been chosen. The algorithm's operation ensures that the % of likelihood of admission is displayed when the user enters information such (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research). The user is given access to a user interface (UI) (Web-based application) where they can enter the above-mentioned information for prediction. The key benefit of this is that the user may use this programme to estimate eligibility and possibility of admission rather than going through the time-consuming procedure of manually determining eligibility for university admission.

1.2. PURPOSE

The goal of this project is to easily estimate an applicant's eligibility for admission to a rated university using a user interface and the given user information (GRE Score, TOEFL Score, University Rating, SOP, LOR, CGPA, Research). Additionally, this removes the chance of human mistake.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEM

Previous studies in this field used the Naive Bayes algorithm to assess the likelihood that a student will be admitted to a particular university, but their main flaw was that they failed to take into account all the variables that would affect admission, such as TOEFL/IELTS, SOP, LOR, and undergraduate GPA. An evaluation network for the applications submitted by university's international students has been built using the Bayesian Networks technique. By comparing prospective students' scores to those of university students currently enrolled, this model was created to predict how well they will do. On the basis of various student scores, the model thus predicted whether the prospective student should be admitted to the university. This method won't be as accurate because comparisons are only made with students who were accepted into the universities, not with those whose admission was denied.

2.2 REFERENCES

- Borah M.D., Application of knowledge based decision technique to predict student enrolment decision, Recent Trends in Information Systems (Re TIS), 21-23 Dec. 2011,180-184.
- Ragab, A.H.M., Hybrid recommender system for predicting college admission, Intelligent Systems Design and Applications (ISDA), 29 Nov. 2012, 107-113
- L. Chang, Applying Data Mining to Predict College Admissions Yield, Chapter 4 in J. Luan and C. Zhao (Eds.), Data mining in action: Case studies, Spring 2008 - College of Education.
- S. Nadi, M.H. Saraee, and A. Bagheri," Hybrid Recommender System for Dynamic Web Users", International Journal Multimedia and Image Processing (IJMIP), Vol. 1, Issue 1, March 2011.

- J. A. Freeman, and D. M. Skapura, “Neural Networks: Algorithms. Applications. And Programming”, AddisonWesley Pub (Sd), June 1991

2.3. PROBLEM STATEMENT DEFINITION

Finding an acceptable college for their future study is the students' biggest challenge. For many students, choosing which institution to apply to is a difficult decision. Our method proposes universities in the order of the student's selection after comparing the student's information with historical admissions data.

3. IDEATION AND PROPOSED SOLUTION

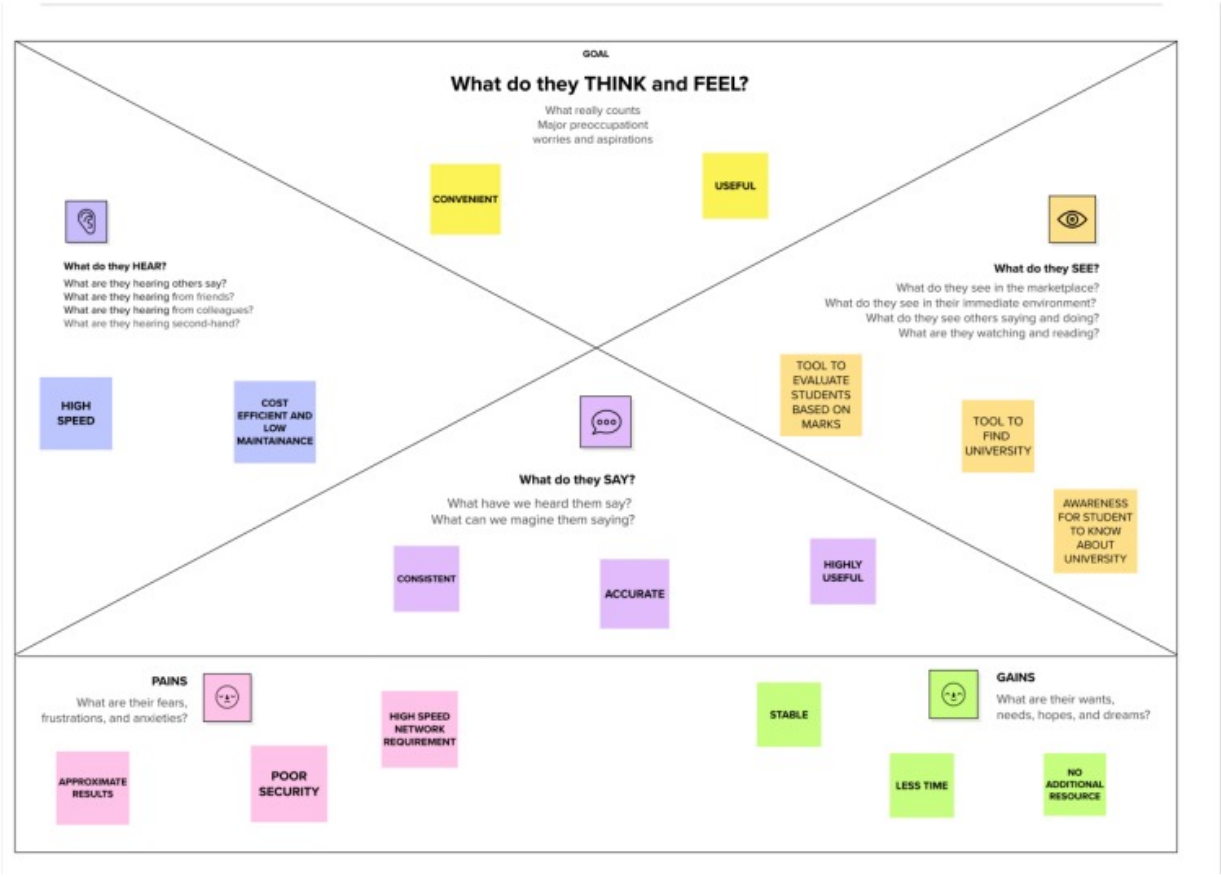
3.1. EMPATHY MAP CANVAS

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to

1) Create a shared understanding of user needs, and

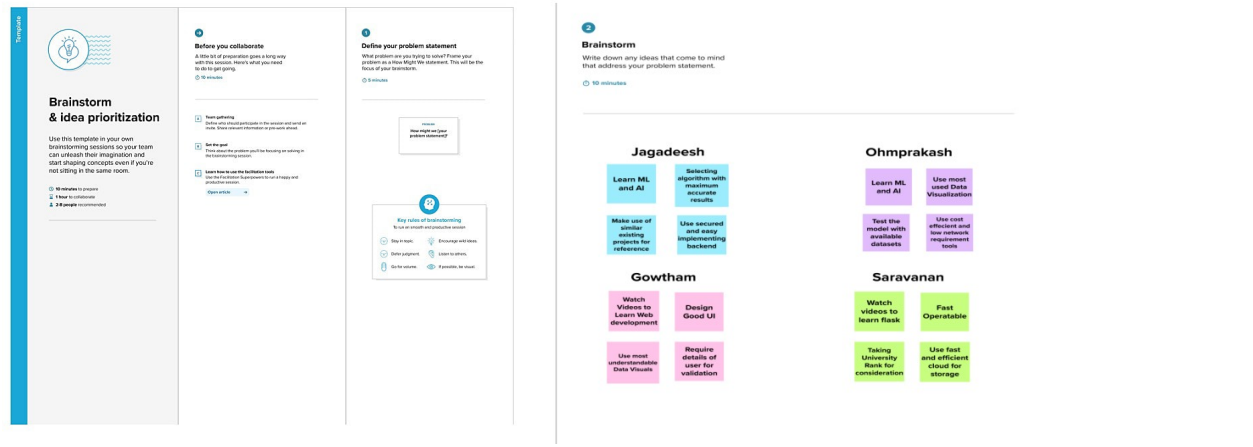
2) Aid in decision making.

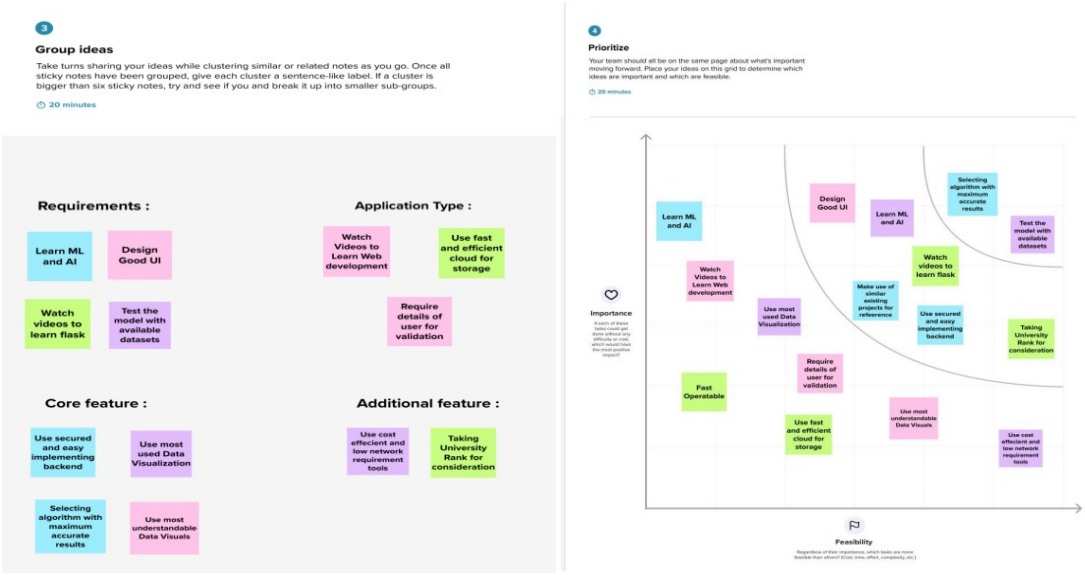
Traditional empathy maps are split into 4 quadrants (Says, Thinks, Does, and Feels), with the user or persona in the middle. Empathy maps provide a glance into who a user is as a whole and are not chronological or sequential.



3.2. IDEATION & BRAINSTORMING

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. In controlled conditions and a free-thinking environment, teams approach a problem by such means as “How Might We” questions. They produce a vast array of ideas and draw links between them to find potential solutions.





3.3. PROPOSED SOLUTION

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system: • Reach to geographically scattered student. • Reducing time in activities • Paperless admission with reduced man power • Operational efficiency

These problems can be resolved by using regression algorithms /classification algorithms as they can consider most of the features for prediction. Linear regression / KNN classification / Random Forest Regressor can be used as the machine learning model for the model. XG boost model can also be used which performs better on small to medium scale datasets but the model giving accurate and desired results only will be selected. The aim of the proposed system is to address the limitations of the current system.

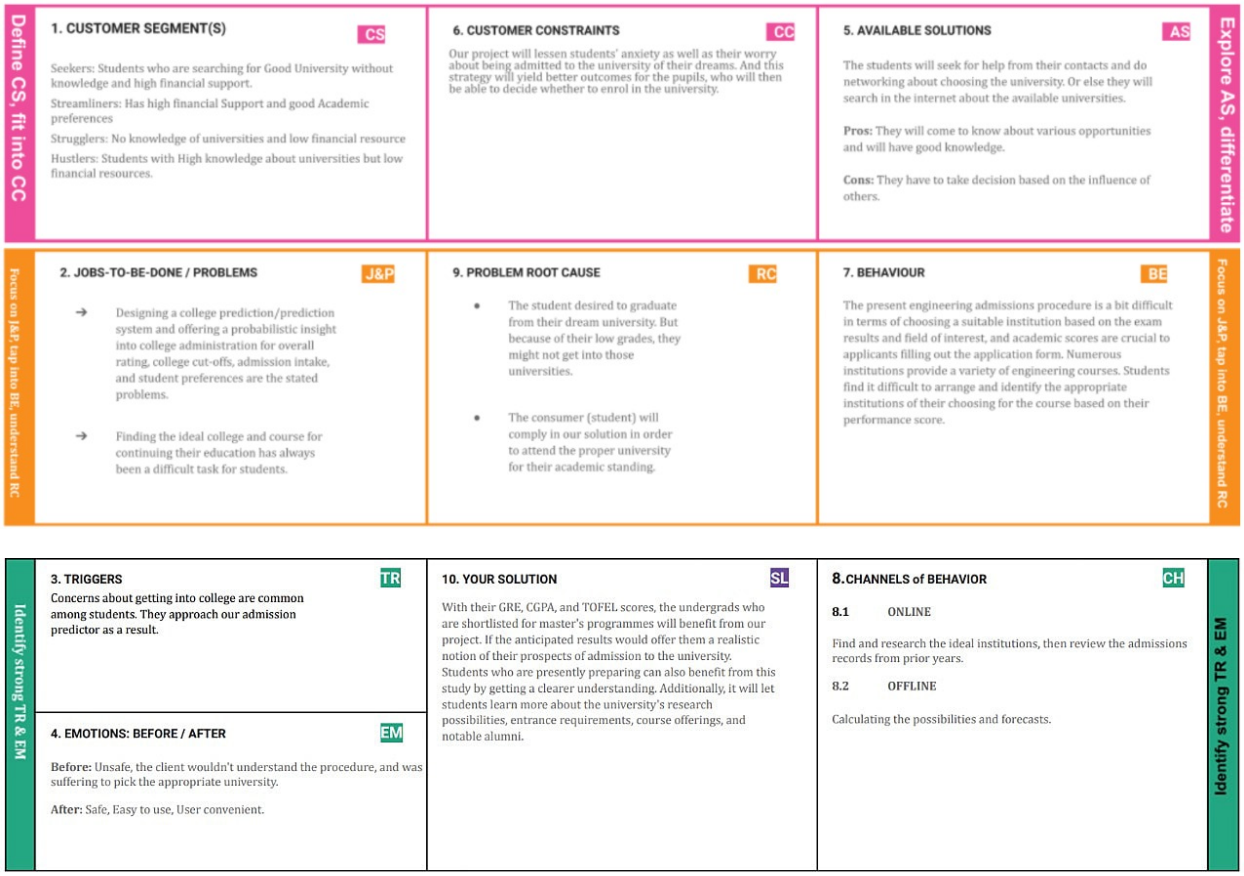
The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools.

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Concerns about getting into college are common among students. This project's goal is to assist students in narrowing down institutions based on their profiles. The anticipated results offer students a good indication of their prospects of admission to a certain university. This analysis should also assist students who are preparing for admission to a master's programme at a university or who will be prepared to do so.
2.	Idea / Solution description	With their GRE, CGPA, and TOFEL scores, the undergrads who are shortlisted for master's programmes will benefit from our project. If the anticipated results would offer them a realistic notion of their prospects of admission to the university. Students who are presently preparing can also benefit from this analysis in order to have a better understanding. Additionally, it will let students learn more about the university's research possibilities, entrance requirements, course offerings, and notable alumni.
3.	Novelty / Uniqueness	The project website can include a summary of the many amenities offered by the institutions as well as directions to get there. Obtain financial aid and scholarship opportunities as well.
4.	Social Impact / Customer Satisfaction	This method will lessen students' anxiety as well as their worry about being admitted to the university of their dreams. Then this The better scores for the pupils will determine whether they are admitted to the institution or not.

5.	Business Model (Revenue Model)	Additionally, marketing the GRE/TOEFL coaching facilities might bring in money. And the University will pay for the website's upkeep and development.
6.	Scalability of the Solution	A conversation room with candidates, instructors, current students, and alumni will be available in a future update. It can be scaled for universities anywhere.

3.4. PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.



4. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these

are captured in use cases.

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
FR-2	User Confirmation	● Confirmation via Email ● Confirmation via OTP
FR-3	User Details	● Enter user details ● Upload proof documents
FR-4	User Requirement	● Upload SSLC and HSLC documents ● Upload Marksheets ● All possible universities for the student can be listed after analysing.

4.2. NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours

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

FR No.	**Non-Functional Requirement**	**Description**

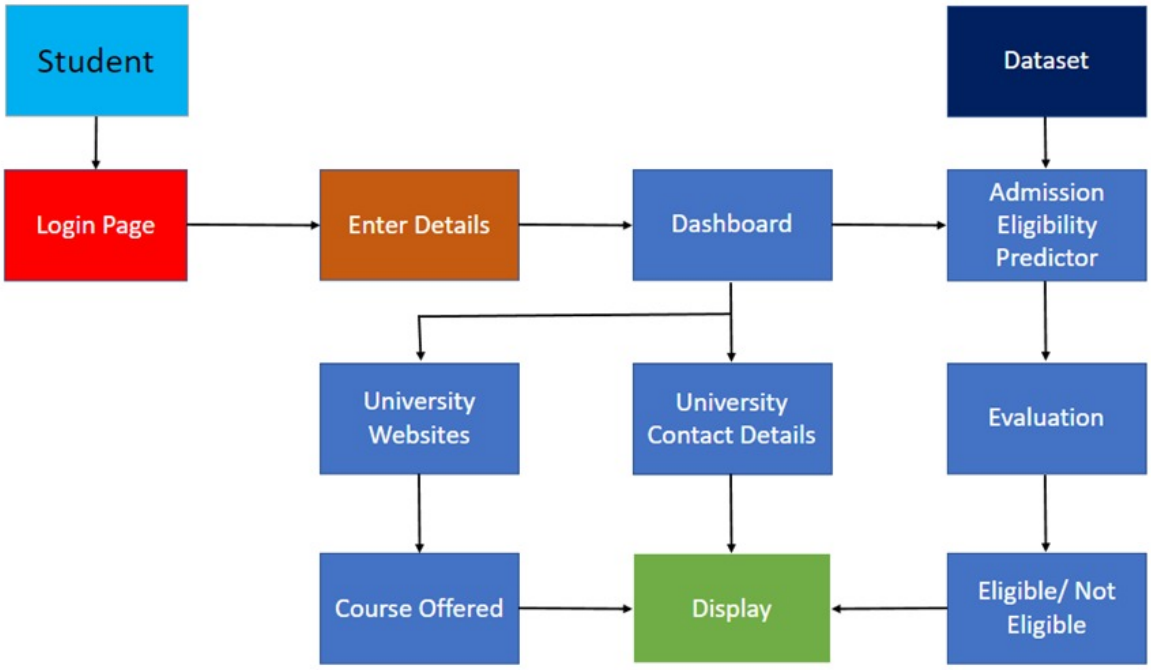
NFR-1	**Usability**	● Good UI. ● User friendly. ● Fast detail fetching, analysing and displaying capability. ● Technical Pre-requisite not required.
NFR-2	**Security**	● Only authenticated user can able to view and edit the user data.
NFR-3	**Reliability**	● The system would always strive for maximum reliability due to the importance of data and damages that could be cause by incomplete and incorrect data
NFR-4	**Performance**	● Database should be backed up every hour ● Under any error, the system should be able to come back to normal operation as soon as possible.
NFR-5	**Availability**	● Less prone to errors ● The system will run 7 days a week, 24 hours a day

NFR-6	**Scalability**	• The admission season is probably when the system will be under the most Strain. So, it must be able to manage numerous concurrent users.

5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAMS

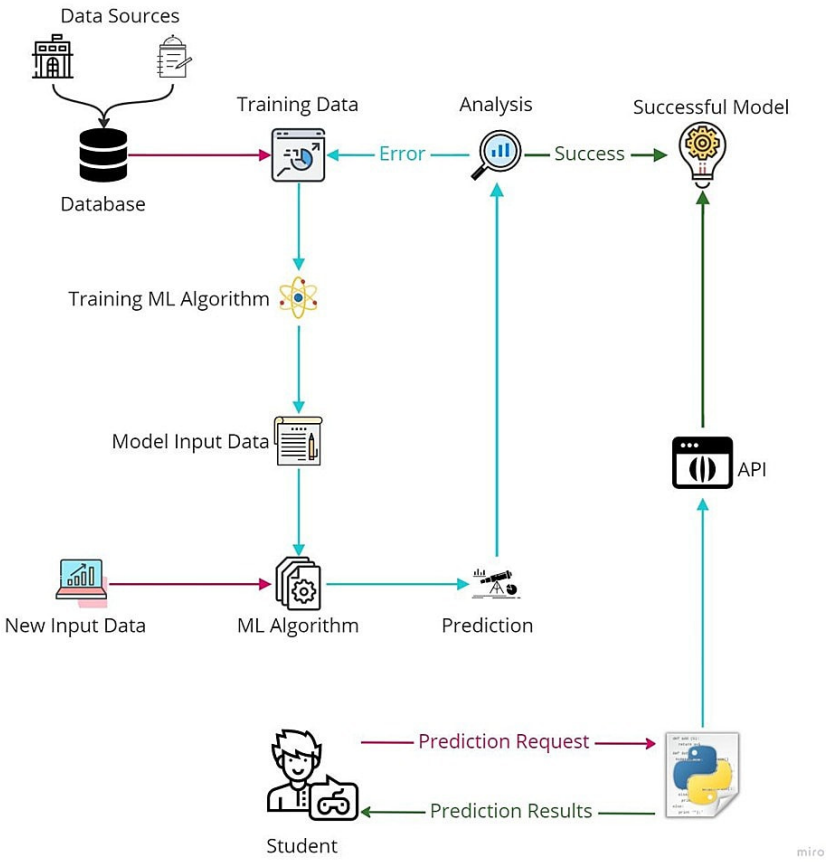
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2. SOLUTION & TECHNICAL ARCHITECTURE

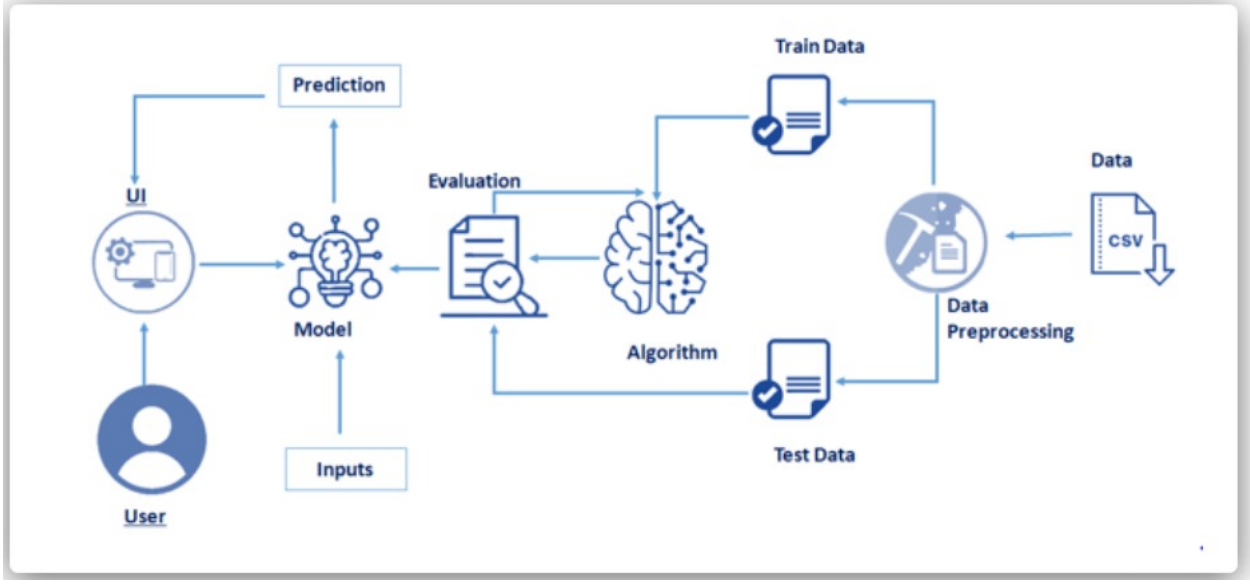
Solution Architecture:

A solution architecture (SA) is an architectural description of a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).



Technical Architecture:

Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that systemrelevant requirements are met.



5.3. USER STORIES

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Login	USN-1	As a user, I can login to the site.	I can enter my login details.	High	Sprint-1
		USN-2	As a user, I can navigate to Form page	I can enter my personnel details, scores and preferred universities.	High	Sprint-1
		USN-3	As a user, I can navigate to dashboard.	I can view the dashboard after entering my details.	High	Sprint-2
	Dashboard	USN-4	As a user, I can view the dashboard.	I can view the university details.	Medium	Sprint-1
		USN-5	As a user, I can see testimonials of students who graduated from the	I can access the Testimonials.	Medium	Sprint-1

			University.			
	Admissions	USN-6	As a user, I can see the previous year cut- off marks	I can download the previous year cut-off details	High	Sprint-2
		USN-7	As a user, I can view review of alumni of the university.	I can access the details of alumni of the university	Medium	Sprint-2
		USN-8	As a user, I can predict my eligibility for admission at the university	I can get result as either eligible/noneligible	High	Sprint-2
	Course Offered	USN-9	As a user, I can see the courses offered by the university for PG students	I can access the course details	Medium	Sprint-2
	Events	USN-10	As a user, I can check various technical events about to happen in the university	I can register for the events	Low	Sprint-3

Administrator	Dashboard	USN-11	As an administratOr, I can update the details about the university	I can check if the update is reflected or not	Medium	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1. SPRINT PLANNING & ESTIMATION

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

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	Jagadeesh Kumar R Saravanan P
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Gowtham M Ohmprakash V
Sprint-2		USN-3	As a user, I can register for the application through Gmail	6	Medium	Ohmprakash V

Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	5	High	Saravanan P Gowtham M
Sprint-3	Selection	USN-5	As a user, I can confirm the available college or re-apply to other college	10	Medium	Jagadeesh Kumar R
Sprint-4	Queries	USN-6	As a user, I can ask queries to the system regarding the help / support or technical issues	10	High	Gowtham M
Sprint-1	Authentication	USN-7	As an admin, I can authenticate the login credentials of user	5	High	Ohmprakash V
Sprint-2	Dashboard	USN-8	As an admin, I can verify the details of the user	7	High	Saravanan P
Sprint-2	Prediction	USN-9	As an admin, I can train the user details with ML algorithm	7	High	Gowtham M Ohmprakash V
Sprint-3	Chances	USN-10	As an admin, I can solve the queries of users	10	High	Saravanan P Gowtham M
Sprint-4	Solution	USN-11	As an admin, I can update the university database depends on the user confirmation	10	High	Jagadeesh Kumar R Saravanan P Ohmprakash V

6.2. SPRINT DELIVERY SCHEDULE

A sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

Sprint	Total Story	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)

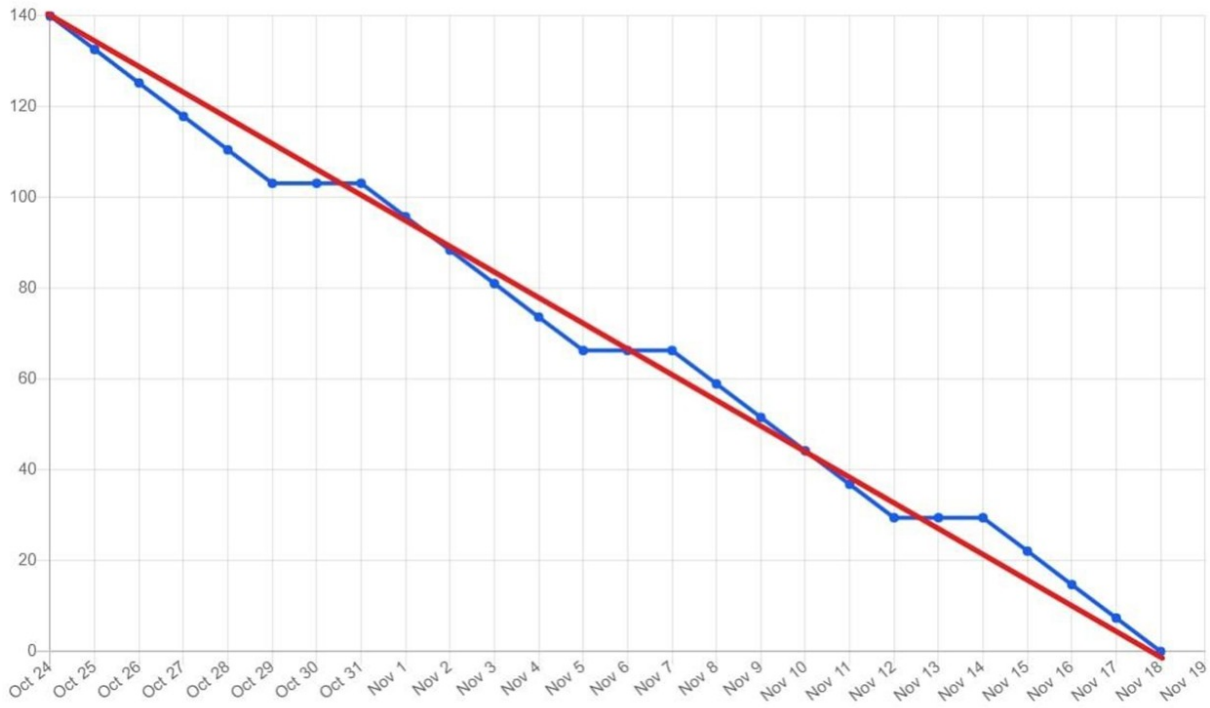
	Points				**(as on Planned End Date)**	
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

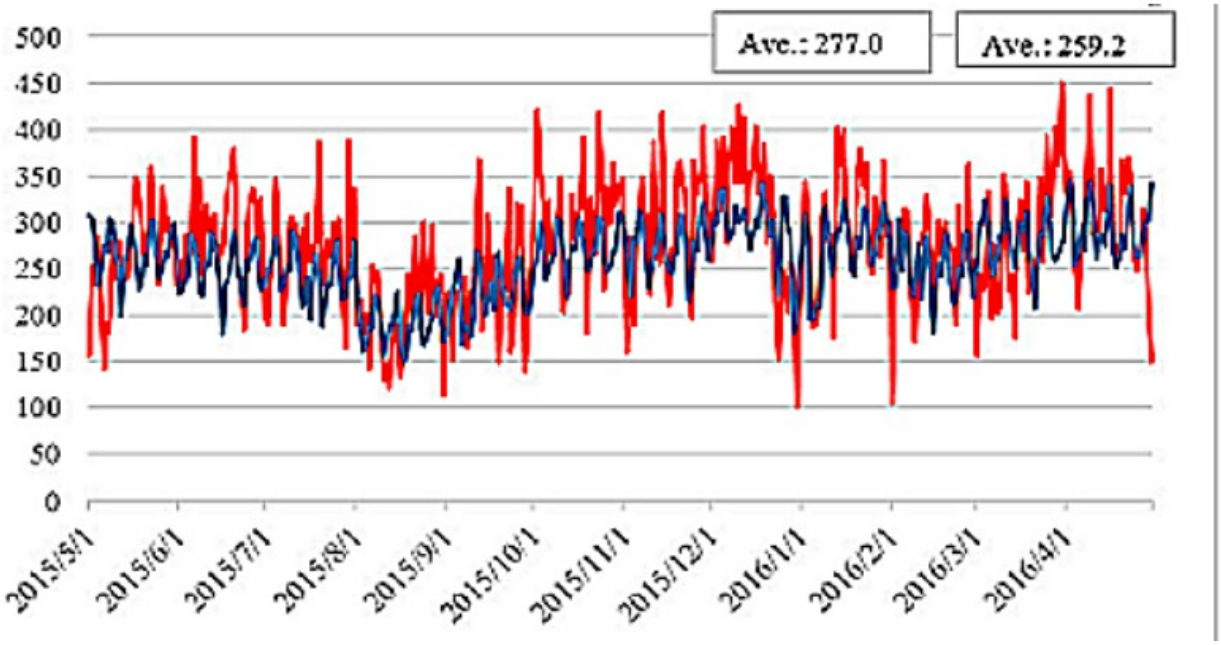
6.3. REPORTS FROM JIRA

Jira helps teams plan, assign, track, report, and manage work and brings teams together for everything from agile software development and customer support to startups and enterprises. Software teams build better with Jira Software, the #1 tool for agile teams.

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.





7. CODING & SOLUITONING

7.1. DATA DICTIONARY

We have used the dataset for building the ML Model and training it. Some of the datas from our dataset are tabulated below.

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4	4.5	8.87	1	0.76
3	316	104	3	3	3.5	8	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.8
5	314	103	2	2	3	8.21	0	0.65
6	330	115	5	4.5	3	9.34	1	0.9

7	321	109	3	3	4	8.2	1	0.75
8	308	101	2	3	4	7.9	0	0.68
9	302	102	1	2	1.5	8	0	0.5
10	323	108	3	3.5	3	8.6	0	0.45

7.2. LIBRARIES USED

- Pandas
- Numpy
- Scikit learn
- Matplotlib
- Seaborn

7.3. TECHNOLOGIES USED

- Software
  - Python
  - Anaconda
  - Jupyter Notebook
  - Windows 11
  - IBM Watson Studio
- Hardware
  - Processor - Quad Core
  - Hard Disk and SSD
  - Memory - 2 GB and Above RAM

7.4. EVALUATION METRIC

The evaluation metric for this competition is 100*RMSLE where RMSLE is Root of Mean Squared Logarithmic Error across all entries in the test set.

7.5. INITIAL APPROACH

- Simple Linear Regression model without any feature engineering and data transformation which gave a RMSE : 196.402.
- Without feature engineering and data transformation, the model did not perform well and

could'nt give a good score.

- Post applying feature engineering and data transformation (log and log1p transformation), Linear Regression model gave a RMSLE score of 0.734.

7.6. ADVANCED MODELS

- With improvised feature engineering, built advanced models using Ensemble techniques and other Regressor algorithms.
- Decision Tree Regressors performed well on the model which gave much reduced RMSLE.
- With proper hyper-parameter tuning, Decision Tree Regressor performed well on the model and gave the lease RMSLE of 0.5237.

8. TESTING

8.1. TEST CASES

Test case ID	Feature Type	Component	Test Scenario
LoginPage_TC 001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button
LoginPage_TC_002	UI	Home Page/Dem02	Verify the UI elements in home page
LoginPage_TC_003	Functional	Chance	Verify that the Candidate Having chance to Admit/Not
LoginPage_TC_004	Functional	NoChance	Verify that the Candidate Having chance to Admit/Not

Execution Steps	**Tested URL**

1. Run the Python Flask code 2. Open your browser 3. Enter the URL and click go 4. Enter the required details 5. Click the predict button	http://127.0.0.1:5000/
6. As per the Student details, the score will be calculated 7. If score is above 50%, then the Chance page will appear	http://127.0.0.1:5000/chance/99.13900931466966
8. If predicted score is below 50%, then the nochance page will appear	http://127.0.0.1:5000/nochance/38.72796854402236

S. No.	Expected Result	Actual Result	Status	Comments
1.	Working as expected	Working as expected	Pass	Perfect Working
2.	Working as expected	Working as expected	Pass	Perfect Working
3.	Working as expected	Working as expected	Pass	Perfect Working
4.	Working as expected	Working as expected	Pass	Perfect Working

8.2. USER ACCEPTANCE TESTING

Defect Analysis:

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

Resolution	Severity1	Severity2	Severity3	Severity4	Sub Total
By Design	7	3	2	3	15
Duplicate	1	0	3	0	4
Extemal	2	3	0	1	6

Fixed	7	3	2	17	29
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	3	2	1	6
Totals	17	12	11	23	63

Test Case Analysis:

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

Section	**TotalCases**	**Not Tested**	**Fail**	**Pass**
PrintEngine	7	0	0	7
ClientAppIication	51	0	0	51
Security	2	0	0	2
OutsourceShipping	3	0	0	3
ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4
VersionControI	2	0	0	2

9. RESULTS

9.1. PERFORMANCE METRICS

S. No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model: MAE -0.03790692243018498, MSE-O.003058753436307664, RMSE — 0.05530599819465936, R2 score — 0.8647260941958439 Classification Model: Confusion Matrix - , Accuray Score- & Classification Report	
2.	Tune the Model	Hyperparameter Tuning: 0.9166666666666666 Validation Method —LinearRegression model	

10. ADVATAGES & DISADVANTAGES

Advantages:

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

Disadvantages:

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

11. APPLICATIONS

- Reach to geographically scattered student.
- Reducing time in activities

- Paperless admission with reduced man power
- Operational efficiency

12. CONCLUSION

The project employs a Random forest regressor to forecast the output, and a web application is created utilising a variety of technologies, including Python, HTML5, CSS, Flask, Scikit, Matplot, Numpy, Pandas, Seaborn, and other libraries, to make the user interface (UI) more accessible and simple. The web application may be viewed from any location with an internet connection after it has been deployed. With this project, you can estimate your eligibility for admission to a ranked university in a fraction of the time.

13. FUTURE SCOPE

Some of the future scopes of this project are:

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

14. APPENDIX

Source Code:

1. HTML Codes:

a. INDEX.HTML:

```
<html>
  <head>
    <title>University Admit Predictor</title>
    <style>
      label{
  display:inline-block;
  width:200px;
  text-align: left;
}
input:not(#research,#noresearch,input[type="submit"]){
  width:350;
  font-size: 17px;
  padding-left:5% ;
```

```
    margin:15 ;
}
.researchlabel{
    margin-top: 10;
    text-align: left;
}
#research,#noresearch{
    margin-top: 7;
    text-align: left;
    margin-left: 20px;
}
.submit{
    margin-top: 30px;
}
.raise:hover,.raise:focus {
  box-shadow: 0 0.5em 0.5em -0.4em var(--hover);
  transform: translateY(-0.25em);
  color: white;
}
.index_body{
    background-image: url('../static/university_bgimg.webp');
    background-repeat: no-repeat;
    background-size: 100% 100%;
    font-size: 18px;
}
.border-cls{
    max-width: fit-content;
    padding: 10px;
    margin-left: 7%;
    width: 60%;
}
p{
    margin-left: 7%;
}
.inline-block{
    display: inline-block;
}
```

```
.gif-outline{
   margin-left: 15%;
}
.img-hor-flip {
   -moz-transform: scaleX(-1);
      -o-transform: scaleX(-1);
      -webkit-transform: scaleX(-1);
      transform: scaleX(-1);
      filter: FlipH;
      -ms-filter: "FlipH";
}
      </style>
   </head>
   <body class="index_body">
      <h1 style="margin-top:40 ;color: blue;text-align: center;">University Admission Prediction System</h1>
      <p style="margin-bottom:30 ;font-size: 27;margin-top: 40;"><b>Enter your details and get probability of your admission</b></p>
      <div class="border-cls inline-block">
         <form action="{{url_for('predict')}}" method="post">
         <label for="GREscore">Enter GRE Score : </label>
         <input type="number" id="GREscore" name="GRE Score"  placeholder="GRE Score (1 to 340)" required="required" min="0" max="340"/>
         <br>
         <label for="TOEFLscore">Enter TOEFL Score : </label>
         <input type="number" id="TOEFLscore" name="TOEFL Score" placeholder="TOEFL Score (1 to 120)" required="required" min="0" max="120"/>
         <br>
         <label for="rating">Enter University Rating : </label>
         <input type="number" id="rating" name="University Rating" placeholder="University Rating (1 to 5)" required="required" min="1" max="5"/>
         <br>
         <label for="SOP">Enter SOP : </label>
         <input type="number" id="SOP" name="SOP" placeholder="SOP (1 to 5)" required="required" onkeypress="return check(event,value)" step="0.1" min="1" max="5"/>
         <br>
         <label for="LOR">Enter LOR : </label>
```

```
        <input type="number" id="LOR" name="LOR" placeholder="LOR (1 to 5)"
required="required" onkeypress="return check(event,value)" step="0.1" min="1" max="5"/>
        <br>
        <label for="CGPA">Enter CGPA : </label>
        <input type="number" id="CGPA" name="CGPA" placeholder="CGPA (1 to 10)"
required="required" onkeypress="return check(event,value)" step="0.1" min="1"
max="10"/><br>
        <label for="research" class="researchlabel">Research : </label><br>
        <input type="radio" id="research" name="research_radio" value="1">
        <label for="research">Research</label><br>
        <input type="radio" id="noresearch" name="research_radio" value="0">
        <label for="noresearch">No Research</label>
        <div  class="submit">
        <input type="submit" class="raise" style="width: fit-content;padding: 7px;padding-left:
10%;padding-right: 10%;font-size: 17px;margin-left: 30%;border-radius: 8px;background-color:
rgb(247, 161, 0);border-color: orange;" value="Predict"/>
        </div>
      </form>
      </div>
      <div class="inline-block gif-outline">
        <img src="../static/search.webp" height="250px" width="350px">
      </div>
  </body>
</html>
```

b. CHANCE.HTML:

```
<html>
  <head>
    <title>Chance</title>
    <style>
      label{
  display:inline-block;
  width:200px;
  text-align: left;
}
```

```
input:not(#research,#noresearch,input[type="submit"]){
    width:350;
    font-size: 17px;
    padding-left:5% ;
    margin:15 ;
}
.researchlabel{
    margin-top: 10;
    text-align: left;
}
#research,#noresearch{
    margin-top: 7;
    text-align: left;
    margin-left: 20px;
}
.submit{
    margin-top: 30px;
}
.raise:hover,.raise:focus {
  box-shadow: 0 0.5em 0.5em -0.4em var(--hover);
  transform: translateY(-0.25em);
  color: white;
}
.index_body{
    background-image: url('../static/university_bgimg.webp');
    background-repeat: no-repeat;
    background-size: 100% 100%;
    font-size: 18px;
}
.border-cls{
    max-width: fit-content;
    padding: 10px;
    margin-left: 7%;
    width: 60%;
}
p{
    margin-left: 7%;
```

```
}
.inline-block{
    display: inline-block;
}
.gif-outline{
    margin-left: 15%;
}
.img-hor-flip {
    -moz-transform: scaleX(-1);
        -o-transform: scaleX(-1);
        -webkit-transform: scaleX(-1);
        transform: scaleX(-1);
        filter: FlipH;
        -ms-filter: "FlipH";
}
        </style>
    </head>
    <body style="background-color:lightgray ;">
        <div>
            <div style="text-align:center;margin-top:10%;">
                <img src="../static/thumbsup.webp" height="230px" width="230px" style="border-
radius:30%">
            </div>
            <div style="text-align: center">
                <h1>Prediction Result</h1>
                <div style="display:flex ;width: 60%;margin-left: 20%;">
                <img style="flex:1" src="../static/party-popper-joypixels.gif" width="100px">
                <h2 style="flex:12">Congratulations! You have a chance to get admission</h2>
                <img style="flex:1;" class="img-hor-flip" src="../static/party-popper-joypixels.gif"
width="100px">
                </div>
            </div>
        </div>
    </body>
</html>
```

c. NOCHANCE.HTML:

```
<html>
    <head>
        <title>No Chance</title>
        <style>
            label{
    display:inline-block;
    width:200px;
    text-align: left;
}
input:not(#research,#noresearch,input[type="submit"]){
    width:350;
    font-size: 17px;
    padding-left:5% ;
    margin:15 ;
}
.researchlabel{
    margin-top: 10;
    text-align: left;
}
#research,#noresearch{
    margin-top: 7;
    text-align: left;
    margin-left: 20px;
}
.submit{
    margin-top: 30px;
}
.raise:hover,.raise:focus {
  box-shadow: 0 0.5em 0.5em -0.4em var(--hover);
  transform: translateY(-0.25em);
  color: white;
}
.index_body{
    background-image: url('../static/university_bgimg.webp');
    background-repeat: no-repeat;
```

```
    background-size: 100% 100%;
    font-size: 18px;
}
.border-cls{
    max-width: fit-content;
    padding: 10px;
    margin-left: 7%;
    width: 60%;
}
p{
    margin-left: 7%;
}
.inline-block{
    display: inline-block;
}
.gif-outline{
    margin-left: 15%;
}
.img-hor-flip {
    -moz-transform: scaleX(-1);
        -o-transform: scaleX(-1);
        -webkit-transform: scaleX(-1);
        transform: scaleX(-1);
        filter: FlipH;
        -ms-filter: "FlipH";
}
        </style>
    </head>
    <body style="background-color:lightgray;">
        <div>
            <div style="text-align:center;margin-top:10%;">
                <img src="../static/sad.webp" height="230px" width="230px" style="border-
radius:30%">
            </div>
            <div style="text-align: center">
                <h1>Prediction Result</h1>
                <h2 style="flex:12">Sorry! You don't have enough score<br>You have chance to get
```

```
into other best universities </h2>
            </div>
          </div>
        </div>
    </body>
</html>
```

2. **PYTHON CODE:**
    a. **APP.PY:**

```
import pandas as pd
from flask import Flask,request, jsonify, render_template, redirect, url_for
import requests
import json

API_KEY = "13S6-gvuJHw0EgY7HAmtl8ae5tQlGcbahHYBYAacEOQn"
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}
app = Flask(__name__, template_folder='templates')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'post'])
def predict():
    GRE_Score = int(request.form['GRE Score'])
    TOEFL_Score = int(request.form['TOEFL Score'])
    University_Rating = int(request.form['University Rating'])
    SOP = float(request.form['SOP'])
    LOR = float(request.form['LOR'])
    CGPA = float(request.form['CGPA'])
    Research = int(request.form['research_radio'])
    final_features = [[GRE_Score, TOEFL_Score, University_Rating, SOP, LOR, CGPA]]
```

```
    payload_scoring = {'input_data': [
        {'field': [["GRE Score", "TOEFL Score", "University Rating", "SOP", "LOR ", "CGPA"]],
         'values': final_features}]}
    print("hello")
    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/3ef17bf1-d7c8-475c-88b4-
1216d3e59253/predictions?version=2022-11-17',
        json=payload_scoring,
        headers={'Authorization': 'Bearer ' + mltoken})
    print("scoring response")
    pred = response_scoring.json()
    print(pred)
    output = pred['predictions'][0]['values'][0][0]
    print(output)
    if output > 0.5:
        return redirect(url_for('chance', percent=output * 100))
    else:
        return redirect(url_for('no_chance', percent=output * 100))

@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])

if __name__ == "__main__":
    app.run(debug=True)
```

b. REGRESSION METHODS.IPYNB

Importing the Libraries

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

```
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Load the dataset

```
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='NgS5Cy5_ZLQF6mBGDOhVIA7GfRn5bRxmnryffm-IuADk',
    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 = 'universityadmitpredictor-donotdelete-pr-pkdavbjvhsmouo'
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 )
df = pd.read_csv(body)
df.head()
```

Analyse the data

```
df.head()
df.tail()
```

Drop Serial No. Column

```
df.drop("Serial No.",axis=1,inplace=True)
df.head()
```

Checking for Null values

```
df.isnull().sum()
```

Getting Information about dataframe

```
df.info()
```

Statistical Summary of Dataframe

```
df.describe()
```

To find correlation of columns

```
corr_matrix=df.corr()
corr_matrix
```

Correlation matrix as a heatmap

```
fig = plt.figure(figsize=(12,8))
sns.heatmap(corr_matrix,annot=True)
plt.show()
```

Data Visualization

Univariant Analysis

```
sns.distplot(df["GRE Score"])
sns.distplot(df["TOEFL Score"])
sns.distplot(df["University Rating"])
sns.distplot(df["SOP"])
sns.distplot(df["LOR "])
sns.distplot(df["CGPA"])
sns.distplot(df["Research"])
sns.distplot(df["Chance of Admit "])
```

Bivariant Analysis

```
sns.relplot(data=df,x="GRE Score",y="Chance of Admit ",hue="Research")
plt.title("GRE Score vs Chance of Admit")
plt.show()
sns.relplot(data=df,x="TOEFL Score",y="Chance of Admit
",hue="Research",kind="line",ci=None)
plt.title("TOEFL vs Chance of Admit")
plt.show()
sns.relplot(data=df,x="CGPA",y="Chance of Admit ",hue="Research")
```

```
plt.title("GRE Score vs Chance of Admit")
plt.show()
sns.relplot(data=df,x="SOP",y="Chance of Admit ",hue="Research",kind="line",ci=None)
plt.title("GRE Score vs Chance of Admit")
plt.show()
sns.relplot(data=df,x="LOR ",y="Chance of Admit ",hue="Research",kind="line",ci=None)
plt.title("GRE Score vs Chance of Admit")
plt.show()
sns.barplot(data=df,x="University Rating",y="Chance of Admit ")
plt.title("University Rating vs Chance of Admit")
plt.show()
df.hist(bins = 30, figsize = (20,20), color = 'blue')
```

Importing the required Libraries for regression model

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
```

Splitting dataset into dependent and independent columns

```
x = df[["GRE Score","TOEFL Score","University Rating","SOP","LOR ","CGPA"]]
y = df["Chance of Admit "]
x.head()
y.head()
```

Splitting dataset into training and testing data

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state=1)
```

Multiple Linear Regression

```
multiple_lin_reg = LinearRegression()
multiple_lin_reg.fit(x_train,y_train)
y_pred_mlr = multiple_lin_reg.predict(x_test)
r2_score_mlr = r2_score(y_test,y_pred_mlr)
print("Mutiple Linear Regression's Score = {:.3f}".format(r2_score_mlr))
```

Random forest regression

```
ran_for_reg = RandomForestRegressor(n_estimators=100,random_state=1)
ran_for_reg.fit(x_train,y_train)
y_pred_rfr = ran_for_reg.predict(x_test)
r2_score_rfr = r2_score(y_test,y_pred_rfr)
print("Random Forest Regression's Score = {:.3f}".format(r2_score_rfr))
```

Save the model

```
import pickle
pickle.dump(multiple_lin_reg,open("model.pkl","wb"))
!pip install ibm_watson_machine_learning
from ibm_watson_machine_learning import APIClient
wml_credentails={
   "url":"https://us-south.ml.cloud.ibm.com",
   "apikey":"13S6-gvuJHw0EgY7HAmtl8ae5tQlGcbahHYBYAacEOQn"
}
client=APIClient(wml_credentails)
def guid_from_space_name(client,space_name):
   space=client.spaces.get_details()
   return(next(item for item in space['resources'] if
item['entity']["name"]==space_name)['metadata']['id'])
space_uid=guid_from_space_name(client,'models')
print("Space UID = "+space_uid)
client.set.default_space(space_uid)
client.software_specifications.list()
software_spec_uid=client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
software_spec_uid
model_details=client.repository.store_model(model=multiple_lin_reg,meta_props={
   client.repository.ModelMetaNames.NAME:"Multiple_Linear_Regression",
   client.repository.ModelMetaNames.TYPE:'scikit-learn_1.0',
   client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid,
},
training_data=x_train,
training_target=y_train
            )
model_details
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-10758-1659202034

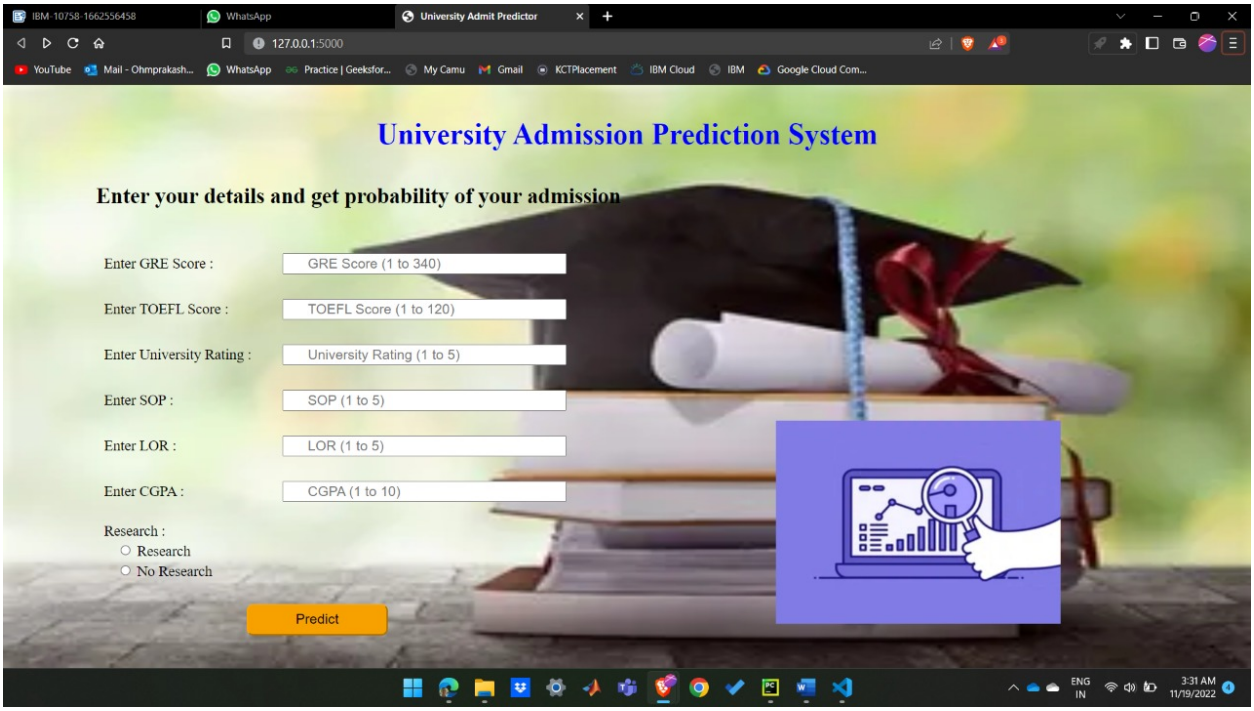
Project Demo Video Link:

https://drive.google.com/file/d/1wkEg34DJyhkRcSDFr3vzNoMRnZocU93z/view?usp=share_link

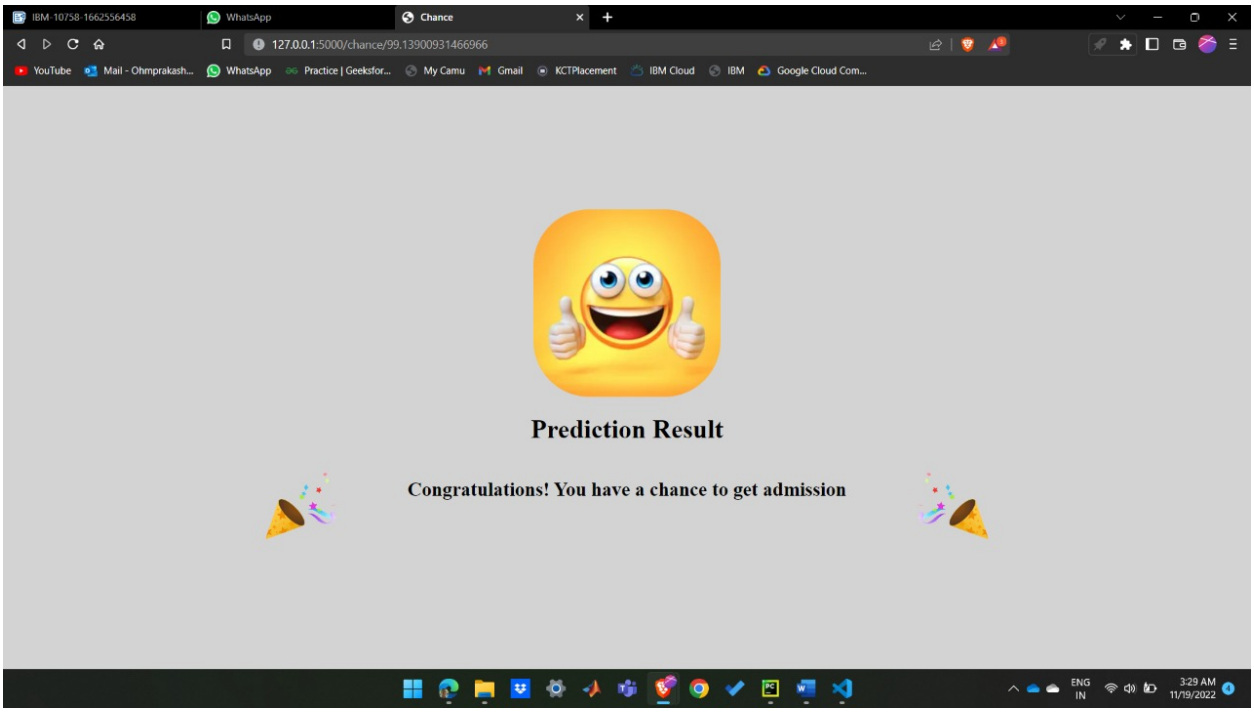
https://www.youtube.com/embed/J1seZumhS_Y

Output Screenshots:

Index Page



Chance Page



No Chance Page