UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID: PNT2022TMID34862

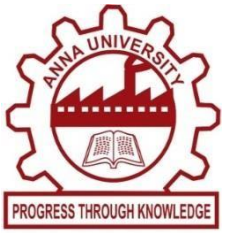
A PROJECT REPORT

Submitted by

MATHAN R (962819104061)
RAJAMANI S (962819104068)
VIGNESH K (962819104087)
VISHWA B (962819104092)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL
ANNA UNIVERSITY::CHENNAI 600 025

NOVEMBER 2022

INDEX

CHAPTER	TITLE	PAGE NO
1.	**INTRODUCTION**	**1**
	1.1 Project Overview	1
	1.2 Purpose	1
2.	**LITERATURE SURVEY**	**2**
	2.1 Existing System	2
	2.2 References	5
	2.3 Problem Statement Definition	6
3.	**IDEATION & PRPOSED SOLUTION**	**7**
	3.1 Empathy Map Canvas	7
	3.2 Ideation & Brainstorming	8
	3.3 Proposed Solution	8
	3.4 Problem Solution Fit	10
4.	**REQUIREMENT ANALYSIS**	**12**
	4.1 Functional Requirement	12
	4.2 Non-Functional Requirements	12
5.	**PROJECT DESIGN**	**15**
	5.1 Data Flow Diagrams	15
	5.2 Solution & Technical Architecture	15
	5.3 User Stories	18
6.	**PROJECT PLANNING &SCHEDULING**	**20**
	6.1 Sprint Planning& Estimation	20
	6.2 Sprint Delivery Schedule	21
	6.3 Reports From JIRA	23
7.	**CODING & SOLUTIONING**	**25**
	7.1 Feature-1	25
	7.2 Feature-2	27

8.	**TESTING**	**31**
	8.1 Test Cases	31
	8.2 User Acceptance Testing	33
9.	**RESULTS**	**35**
	9.1 Performance Metrics	35
10.	**ADVANTAGES & DISADVANTAGES**	**36**
11.	**CONCLUSION**	**37**
12.	**FUTURE SCOPE**	**38**
	APPENDIX	**38**
	Source Code	**47**
	GitHub & Project Demo Link	**47**

ABSTRACT

Many students currently pursue their education outside of their home nations. These international students mostly have the United States of America in mind. India and China account for the majority of foreign students in the United States of America. The number of Indian students enrolling in postgraduate programmes in the USA has sharply increased during the last ten years. Each applicant must contend with a challenging admission process due to the rise in the number of overseas students studying in the United States. It might be challenging for aspiring graduate students to decide which universities to apply to. Students frequently question whether their resume is strong enough for a particular university. This issue has been dealt with in this research by modelling a recommender system based on different classification techniques. Thegradcafe.com provided the necessary data.

Based on this data set, several models were developed, and the best one was selected to recommend universities to the students along with others having comparable features. This increased the likelihood that the student would be admitted from that list of universities. Classification algorithms have also been used to forecast a student's likelihood of admission to a specific university. 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 is designed to display the % of possibility of admission when the user enters information such as GRE, TOEFL, SOP, LOR, CGPA, and University Rating. A user interface is given to the user so they can enter the above-mentioned information for prediction.

CHAPTER 1
INTRODUCTION

1.1 Project Overview

Students are often worried about their chances of admission toUniversity. 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.

1.2 Purpose

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

CHAPTER 2
LITERATURE SURVEY

BERAT UJKANI [1]: The Matura exam is the final national examination that high school students in many countries must pass to be eligible for admission to a university. This paper discusses the key factors that have the most impact in producing a reliable machine learning model for predicting students' enrolment in the university. These factors include the final grades from each high school year, Matura exam results and the university entry exam points. It should be noted that demographic factors were not taken into consideration in this study. Four machine learning (ML) techniques with a total of sixteen algorithms were implemented using the Weka software: Bayes (Bayes Net, Naive Bayes, etc.), Logistic Regression (Logistic and Simple Logistic), K-Nearest Neighbours (IBK, K Star, and LWL) and Decision Tree (J48, Random Forest, Rep Tree, etc.). According to the results, The Rep Tree algorithm performed the best with a True Positive (TP) rate of 0.902 and a False Positive (FP) rate of 0.148. The algorithm with the lowest performance was Naïve Multi with a TP rate of only 0.790. However, the range between the worst and the best-performing algorithms was 14.18%.

OMAER FRAUQ GONI[2]: Every year many students apply for graduate admission to different universities. To select an applicant, each university has different selection criteria such as GRE score, CGPA, research background, statement of purpose, letter of recommendation, university rating etc. There are some web applications as well as some consultancy services for suggesting the appropriate university based on students' portfolio. These help to give an idea which universities should be applied for admission. But they have limitations because humans are incapable of considering all the conditions and universities. Moreover, web applications have accuracy problems. In this study, we have proposed a deep neural network (DNN) to predict the chance of

getting admitted to a university according to the students portfolio. All the selection criteria are considered here to predict the chance of admission. The DNN model has been compared with existing methods in terms of different performance metrics including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), R-squared score. It has shown the most promising result that includes R-squared score of 0.8538 and MSE of 0.0031. The proposed method has also outperformed all the existing methods in each benchmark .

JAYASHREE KATTI[3]: For a pursuing graduate student, shortlisting the colleges could be an intense issue. College undergraduates frequently have an inclination to ponder over the chance that their profile suits the college requirements. Computer programs are exceptionally well trained and faster than humans in making decisions. Moreover, the cost of admission in a college is a lot, making it very crucial for a student that their profile gets shortlisted for a university admission. A University prediction machine learning algorithm is very advantageous for college undergraduates to choose their dream university which also matches their resume. The proposed method considers diverse variables related to the student and his score in various tests. The dataset includes LOR, GRE score, CGPA, TOEFL score, University rating, SOP, etc. Based on all these criterias, the admission to a particular university of an undergraduate will be predicted.

ABDUL HAMID M RAGAB [4]: This paper presents a new college admission system using hybrid recommender based on data mining techniques and knowledge discovery rules, for tackling college admissions prediction problems. This is due to the huge numbers of students required to attend university colleges every year. The proposed HRSPCA system consists of two cascaded hybrid recommenders working together with the help of college predictor, for achieving high performance. The first recommender assigns student's tracks for preparatory year students. While the second recommender

assigns the specialized college for students who passed the preparatory year exams successfully. The college predictor algorithm uses historical colleges GPA students admission data for predicting most probable colleges. The system analyzes student academic merits, background, student records, and the college admission criteria. Then, it predicts the likelihood university college that a student may enter. A prototype system is implemented and tested with live data available in the On Demand University Services (ODUS) database resources, at King Abdulaziz University (KAU). In addition to the high prediction accuracy rate, flexibility is an advantage, as the system can predict suitable colleges that match the students' profiles and the suitable track channels through which the students are advised to enter. The system is adaptive, since it can be tuned up with other decision makers attributes performing trusted needed tasks faster and fairly.

HANAN ABDULLAH MENGASH[5]: An admissions system based on valid and reliable admissions criteria is very important to select candidates likely to perform well academically at institutions of higher education. This study focuses on ways to support universities in admissions decision making using data mining techniques to predict applicants' academic performance at university. A data set of 2,039 students enrolled in a Computer Science and Information College of a Saudi public university from 2016 to 2019 was used to validate the proposed methodology. The results demonstrate that applicants' early university performance can be predicted before admission based on certain pre-admission criteria (high school grade average, Scholastic Achievement Admission Test score, and General Aptitude Test score). The results also show that Scholastic Achievement Admission Test score is the pre-admission criterion that most accurately predicts future student performance. Therefore, this score should be assigned more weight in admissions systems. We also found that the Artificial Neural Network technique has an accuracy rate above 79%, making it superior to other classification techniques.

A.Sivasangari[6]: In the present conditions, students regularly have difficulty finding a fitting institution to pursue higher studies based on their profile. There are some advisory administrations and online apps that recommend universities but they ask huge consultancy fees and online apps are not accurate. So, the aim of this research is to develop a model that predict the percentage of chances into the university accurately. This model provides also the analysis of scores versus chance of prediction based on historical data so that students can understand whether their profile is suitable or not. The proposed model uses linear regression and random forest algorithms but cat boost algorithm is giving highest accuracy.

2.1 Existing problem

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

2.2 References

1. Geiser, Saul, and with Roger Studley. "UC and the SAT: Predictive validity and differential impact of the SAT I and SAT II at the University of California." Educational Assessment
8.1 (2002): 1-26.

2. Rothstein, Jesse M. "College performance predictions and the SAT." Journal of Econometrics 121.1-2 (2004): 297-317.
3. Leonard, David K., and Jiming Jiang. "Gender bias and the college predictions of the SATs: A cry of despair." Research in Higher education 40.4 (1999): 375-407.

2.3 Problem Statement Definition

Problem Statement:

➢ The problem includes the questions like whether a student will get an admit or not?

➢ What are the parameters for selection in the university?

➢ What is the probability of liking selected by the university? And can it be mathematically expressed?

➢ In this case, our objective is to predict whether a student will get an admit or not? and the probability of getting admitted based on selection criteria.

Problem Solution:

➢ For this Problems, all University eligibility parameters should be taken into consideration, such as the students GPA, students Rank, on the other entrance exam scores such as ACT reading score, writing score, source score and English score. (Entrance exams may differ for different universities).

➢ Based on the entered parameters, the model compares it with historical admission data of the selected university.

➢ This data is very useful in analysing whether the student with can get admitted into the university.

➢ Now the probability of getting admitted into the university can be protected using various regression algorithms such as linear regression, logistic regression, decision tree regression.

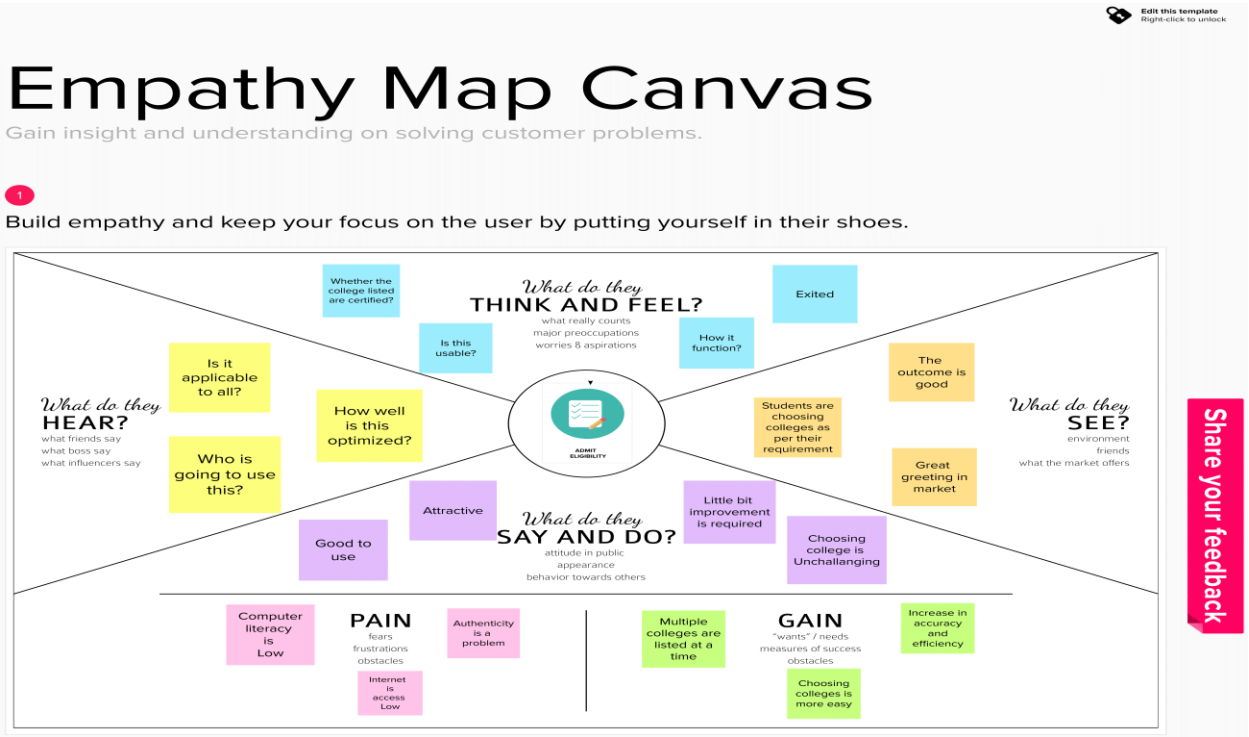
CHAPTER 3

IDEATION & PROPOSED SOLUTION

The project aims to develop an application that uses artificial intelligence with the help of chat bot to customize products for the customers which enhances the fame of ecommerce store and reduce the time which customers spends on choosing products. The application also uses IBM cloud storage for storing objects.

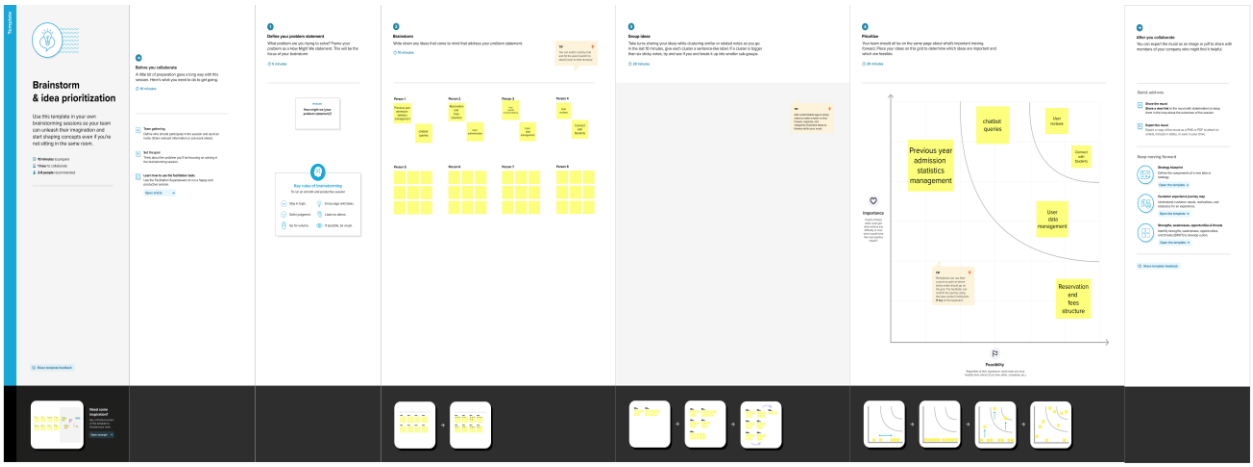
An application that predicts the university admission chances of a student powered by machine learning models. Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. The primary objective of this research is to develop a system to solve the problems the international students are facing while applying for universities in the USA.

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

This task of shortlisting the universities where the student has high chances of admission is difficult for mainly for the international students, so they end up with applying to many universities in hopes of getting admission in few of them thus investing an extra amount of money in the applications. There are several portals and websites which provide information and help to students in shortlisting the universities, but they are not reliable. Most of the students don't take the risk of evaluating the colleges by themselves, and they seek the help of the education consultancy firms to do it for them. Again for this students have to pay a huge amount of fee to the education consultant.



3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The problem statement, hence being tackled, is to design a college prediction/prediction system and to provide a probabilistic insight into college administration for overall rating, cut-offs of the colleges, admission intake and preferences of students.
2.	Idea / Solution description	By using Graduate admission2 dataset in the colleges are predicted from the dataset.
3.	Novelty / Uniqueness	The system provides mock test for the students to choose the course.
4.	Social Impact / Customer Satisfaction	Students are benefited and choosing the college would become effort less.
5.	Business Model (Revenue Model)	It is used among the students and can easily choose their desire a college. And also used in counselling.
6.	Scalability of the Solution	Based on the marks that they obtained in their higher secondary (HSC).

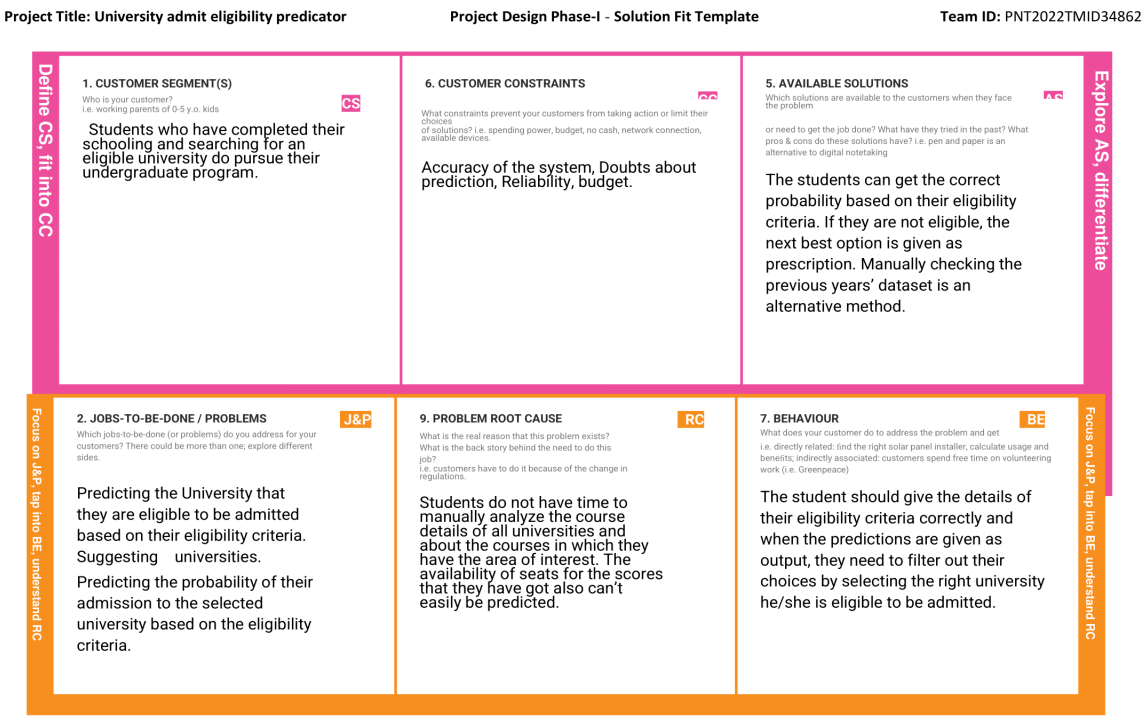
Finally, K Nearest Neighbours and Decision Tree algorithms were used as they were found to be the best fit for the system developed. Also, we will be creating a simple user interface which will help the users to input the data related to student profile and get the predicted result for the application based on the profile as output. This research will thus eventually help students saving the extra amount of time and money they have to spend at the education consultancy firms. And also, it will help them to

limit their number of applications to a small number by proving them the suggestion of the universities where they have the best chance of securing admission thus saving more money on the application fees.

3.4 Problem Solution fit

We will be developing a University Admit Eligibility Predictor system which will help the students to predict the chances of their application being selected for a particular university for which they wish to apply based on their profile. Also, the system will provide a recommendation of universities to the student to which the student has a high possibility of getting admission.

Multiple machine learning classification algorithms were evaluated to develop the system.



Identify strong TR & EM

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news

Students who get to know about the platform from previous users who have got correct predictions. The Search for finding a platform in a browser to predict admit eligibility can also be a driving force.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Confused, uncertain clear, Certain The students get the idea of which university they are eligible to be admitted

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

This project gives the exact probability of getting admitted into the university that the student selects based on their scores and other criteria and also provides them with suggestions based on the same . This effectively decreases the time spent in finding a suitable university.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

8.1 ONLINE

The student can check the university admission criteria online.

8.2 OFFLINE

The student can personally visit the university in person and proceed with the admission process

CHAPTER 4
REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

4.1 Functional requirements

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linked IN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Submit the documents • GRE or/and TOEFL Score Sheet • Curriculum Vitae (CV) • Statement of purpose (SOP) • Le er of Recommendation

FR-4	User Requirements	• Upload all the relevant documents in the appropriate location in the website • Based on the uploads the system would scrape all the necessary information • The List of all possible University for the candidate would be displayed based on the scarped information

4.2 Non-functional Requirements:

NFR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	➢ Good User Interface. ➢ The interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand.
NFR-2	Security	➢ Data inside the system will be protected against malware a acks or unauthorized access.

NFR-3	Reliability	➢ The prediction made by the model is very accurate so that it is always reliable.
NFR-4	Performance	➢ Limited control over third party domains.
NFR-5	Availability	➢ The prediction made by the model is very accurate so that it is always reliable.
NFR-6	Scalability	➢ Able to manage numerous concurrent users.

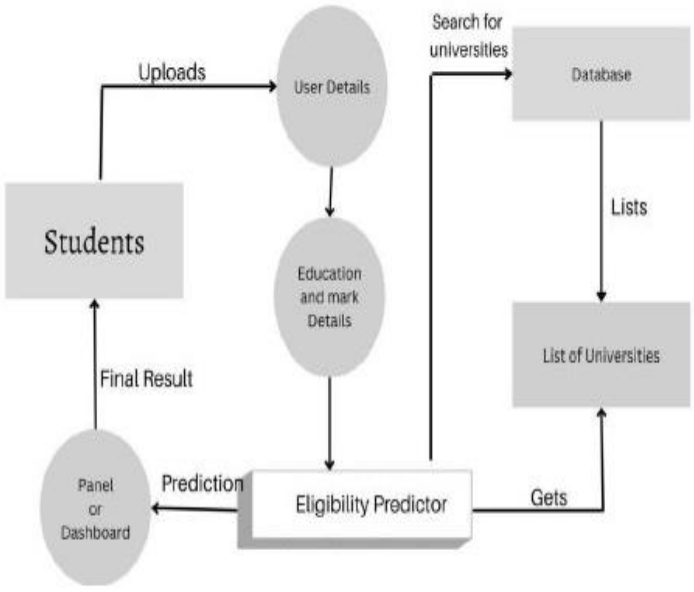
- Speed
- Security
- Portability
- Compatibility
- Capacity
- Reliability
- Environment
- Localization

CHAPTER 5

PROJECT DESIGN

5.1 Data Flow Diagrams

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



5.2 Solution & Technical Architecture

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

1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.

3. Define features, development phases, and solution requirements.

Provide specifications according to which the solution is defined, managed, and delivered.

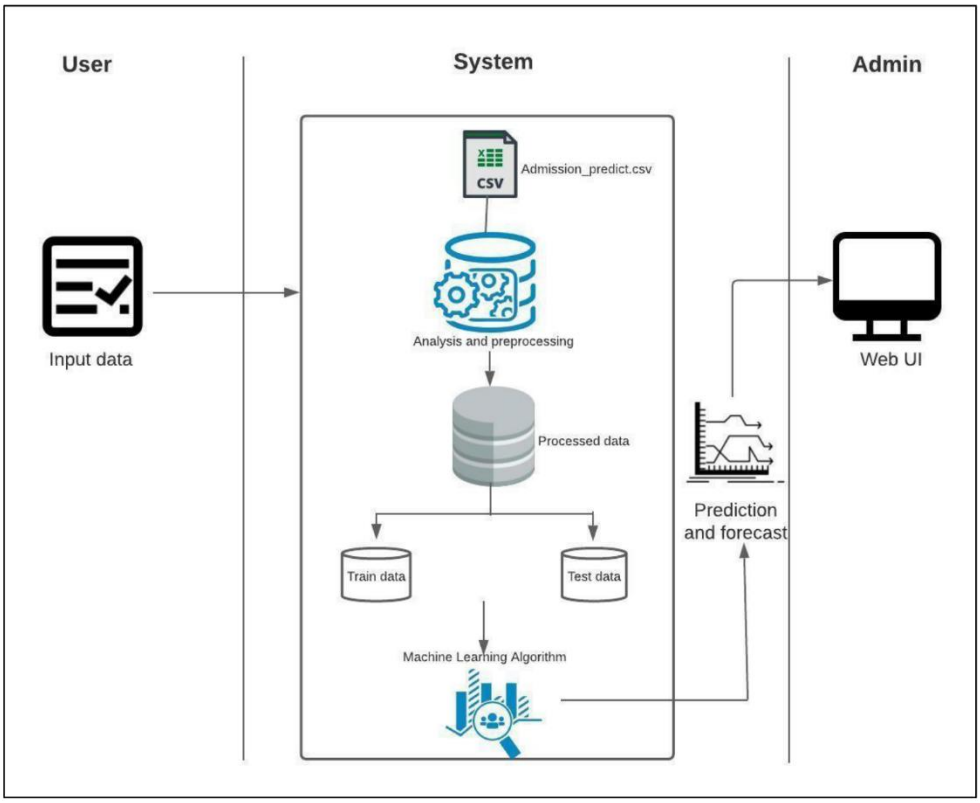


Table-1: Components and Technologies:

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
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud ant etc.
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 Local Server Configuration: Built-in Flask web server	Flask, Web server

Table-2: Application Characteristics:

S.No.	Characteristics	Description	Technology

1.	Open-Source Frameworks	Flask	Micro web framework with python
2.	Security Implementations	Http authentication, Session based authentication	Flask security
3.	Scalable Architecture	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	Flask
4.	Availability	Higher compatibility with latest technologies and allows customization	Flask
5.	Performance	Integrated support for unit testing. 1. RESTful request dispatching. 2. Uses Jinja templating. 3. Support for secure cookies	Flask

5.3 User Stories

A user story is an informal, natural language description of features of a software system. They are written from the perspective of an end user or user of a system, and

may be recorded on index cards, post-it notes, or digitally in project management software.[1] Depending on the project, user stories may be written by different stakeholders like client, user, manager, or development team.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Student)	Dashboard	USN-1	As a user, I can view the cut off marks of previous years in my dashboard	I can access and download the files	High	Sprint-1
		USN-2	As a user, I can view university details and their rankings	I can only view(read-only)	Medium	Sprint-1
		USN-3	As a user, I can review the experience of the students in the university	I can access the review sections	Medium	Sprint-2
		USN-4	As a user, I can upload my documents	I have read and write access to upload files	High	Sprint-1
		USN-5	As a user,I can fill out the general and educational details in the form provided	I have read and write access to the forms filled	High	Sprint-2
	Predictor	USN-6	I can view the list of universities in which I am eligible to get an admission	I can receive the final result as whether eligible or not	High	Sprint-2
		USN-7	I can view the list of universities I am eligible with the same cut-off but in previous years	I can access the files with read-only permission	Medium	Sprint-2
Administrator	Dashboard	USN-8	As an administrator, I can have access to update the latest updates of the universities	I can have access to read and write the university information in the dashboard	High	Sprint-3
		USN-9	As an administrator, I can access any resources available in the page	I can access the resources that are available	Medium	Sprint-3
		USN -10	As an administrator, I can have a track on the universities the student is eligible to get admission at predicted.	I can access the list of the universities obtained as final result	High	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	**Functional**	**User**	**User Story / Task**	**Story**	**Priority**	**Team Members**
	Requirement	**Story**		**Points**		
	(Epic)	**Number**				
Sprint-1	Registration	USN-1	As a user, I will be able to register my application by entering my email, password, and confirming my password.	2	High	Vishwa B
Sprint-2		USN-2	As a user, I will be able to receive an email confirmation after registration.	1	High	Vignesh k
Sprint-2		USN-3	As a user, I can register for the application through Gmail.	2	Low	Vignesh K
Sprint-1		USN-4	As a user, I can register for the application by	2	Medium	Vishwa B

			entering details by self.			
Sprint-3	Data set	USN-6	Add the Data set	1	Low	Mathan R
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Vishwa B
Sprint-3	Html and Css	USN-7	Static and template the use predicts executed	1	High	Mathan R
Sprint-4	Predicted	USN-8	All the process as web created as predicted	1	High	Raja Mani S
Sprint-4	Output	USN-9	Output all predicates	1	High	Raja Mani S

6.2 Sprint Delivery Schedule

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022

Sprint-2	20	6 Days	30 Oct 2022	02 Sept 2022	20	31 Oct 2022
Sprint-3	20	6 Days	01 Sept 2022	07 Sept 2022	20	05 Sept 2022
Sprint-4	20	6 Days	06 Sept 2022	15 Sept 2022	20	12 Sept 2022

6.2.1. Velocity

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

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

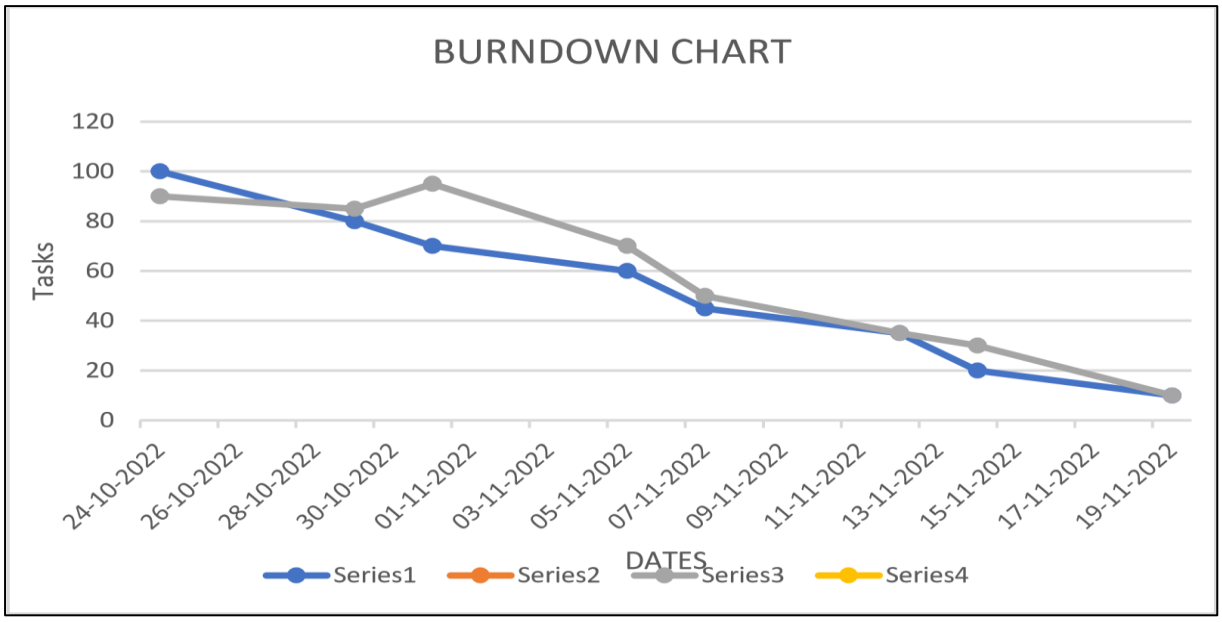
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day).

6.3 Reports from JIRA

6.3.1. Burndown chart:

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

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



CHAPTER 7

CODING & SOLUTIONING

7.1 Feature 1-FLASK APP

Coding:

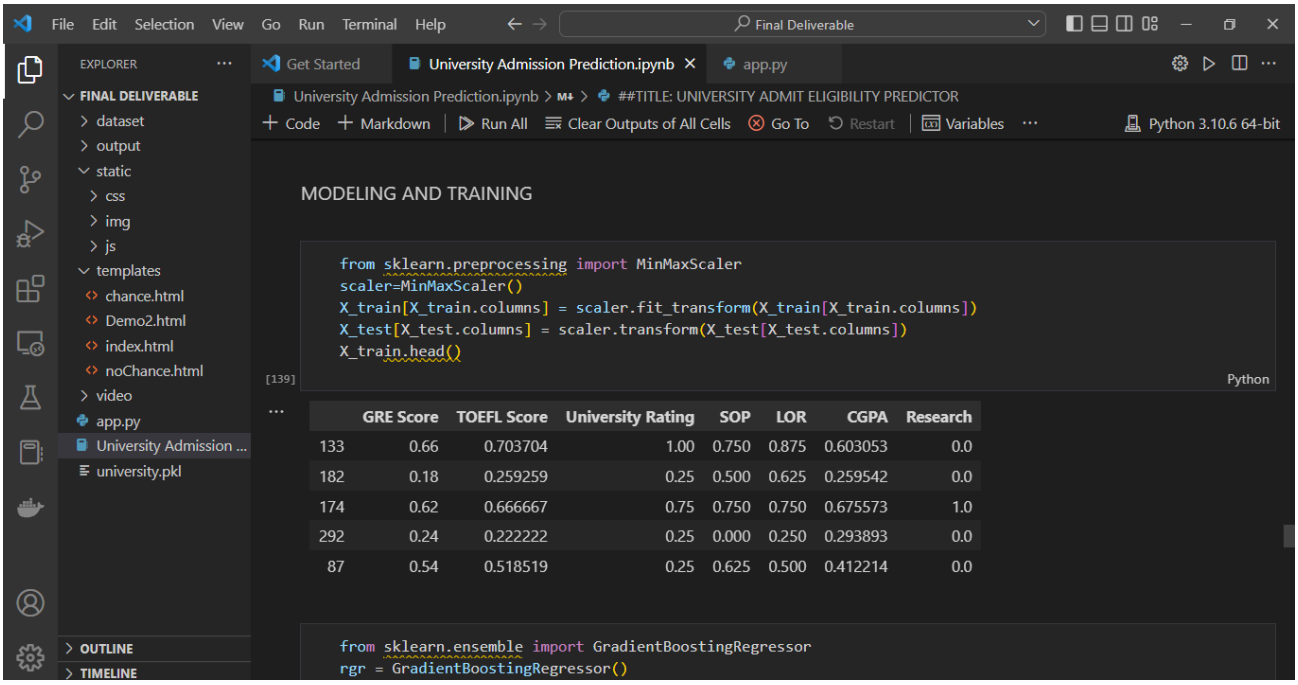
App.py

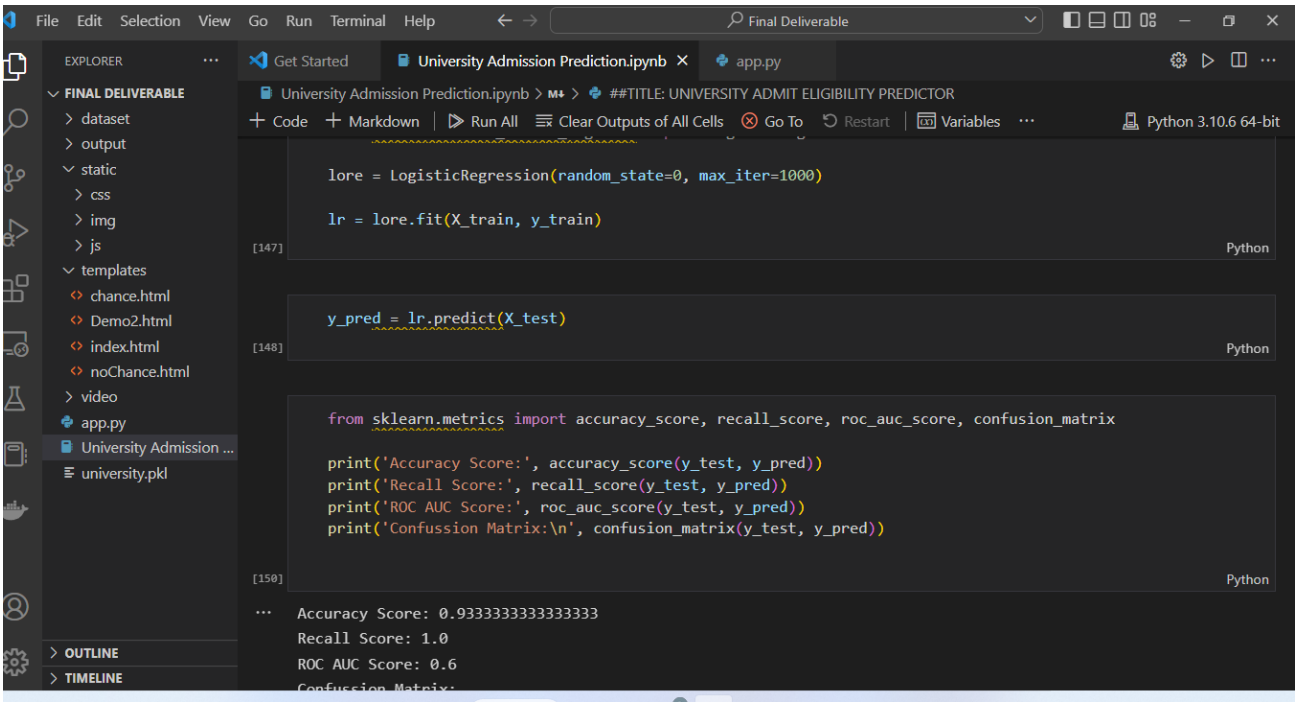
```
from flask import Flask, render_template, redirect, url_for, request
import requests

app = Flask(__name__)

@app.route("/", methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
        arr = []
        for i in request.form:
            val = request.form[i]
            if val == '':
                return redirect(url_for("demo2"))
            arr.append(float(val))

        # deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
        API_KEY = "wf8mge_OQdwVO8ao2kmWCtfxOfLWl8442SH44V85v2Ls"
        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',
```

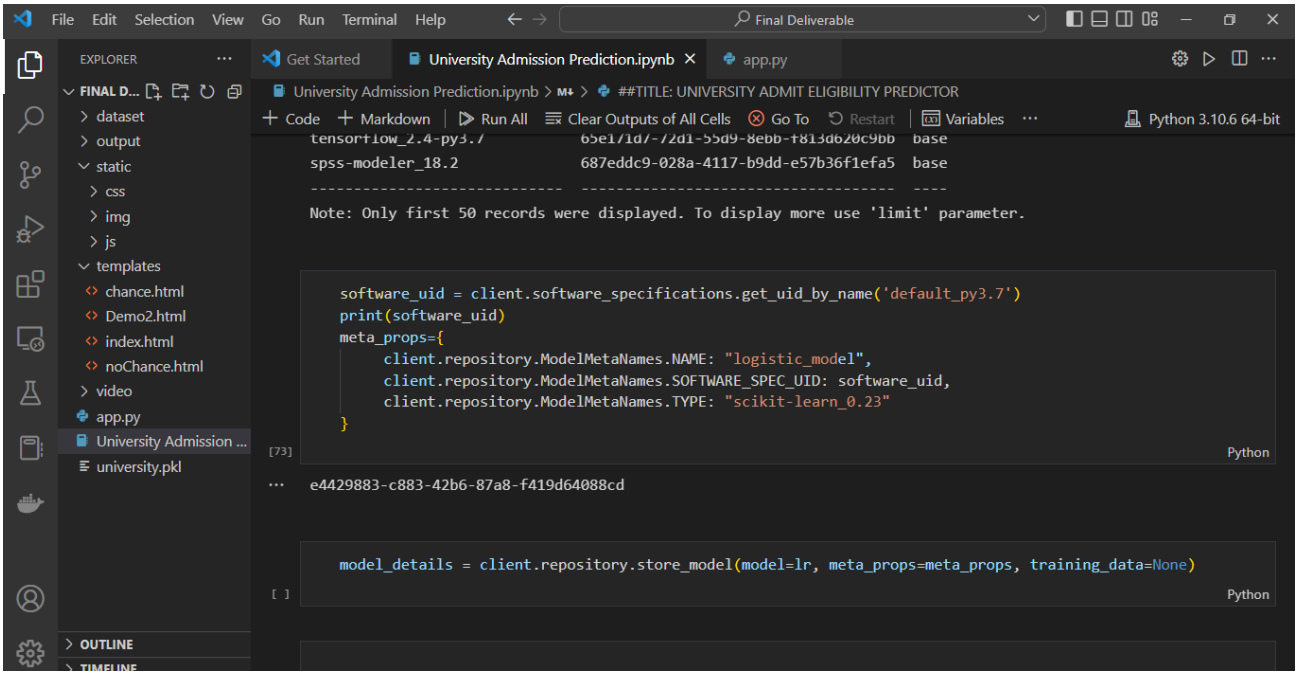
```
                    'TOEFL Score',
                    'University Rating',
                    'SOP',
                    'LOR ',
                    'CGPA',
                    'Research'],
                "values": [arr]
                }]
            }

    response_scoring = requests.post(
        'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/8308fd4c-24a5-46ab-96fa-
263657ae4ad0/predictions?version=2022-10-18',
        json=payload_scoring,
        headers=header
    ).json()

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

    if result[0][0] > 0.5:
        return redirect(url_for('chance', percent=result[0][0]*100))
    else:
        return redirect(url_for('no_chance', percent=result[0][0]*100))
  else:
    return redirect(url_for("demo2"))

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

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

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

@app.route("/nochance/<percent>")
def no_chance(percent):
    return render_template("noChance.html", content=[percent])

@app.route('/<path:path>')
def catch_all():
    return redirect(url_for("demo2"))

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

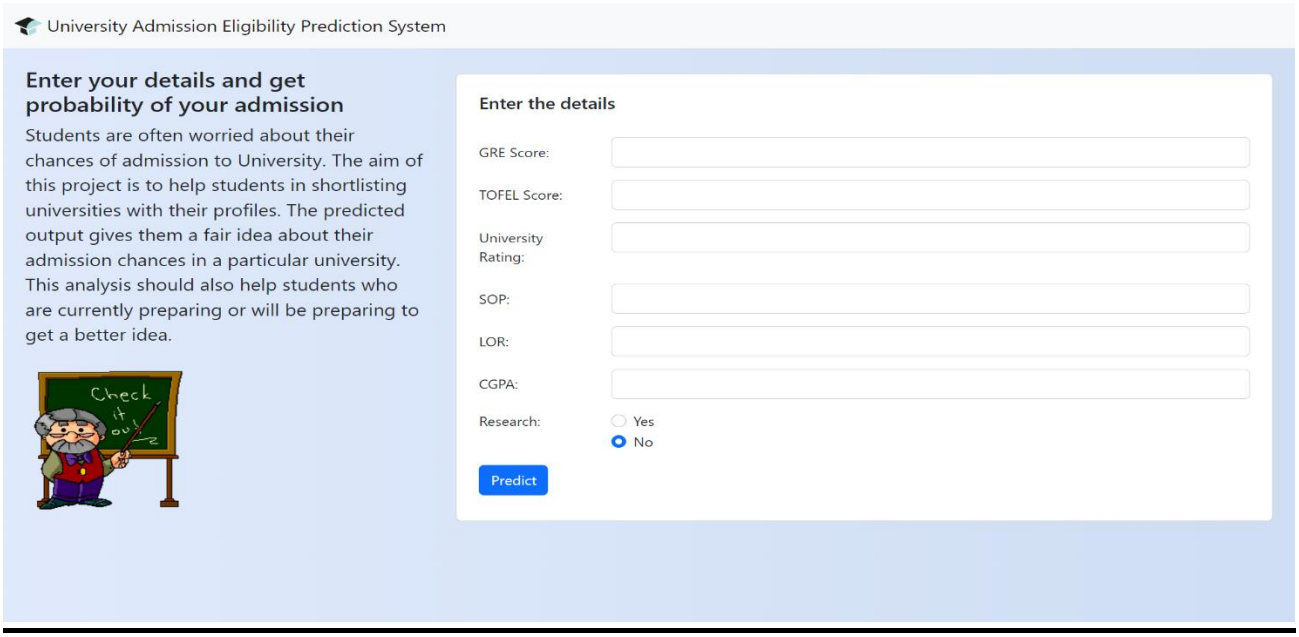
7.2 Feature 2- IBM CLOUD DEPLOYED

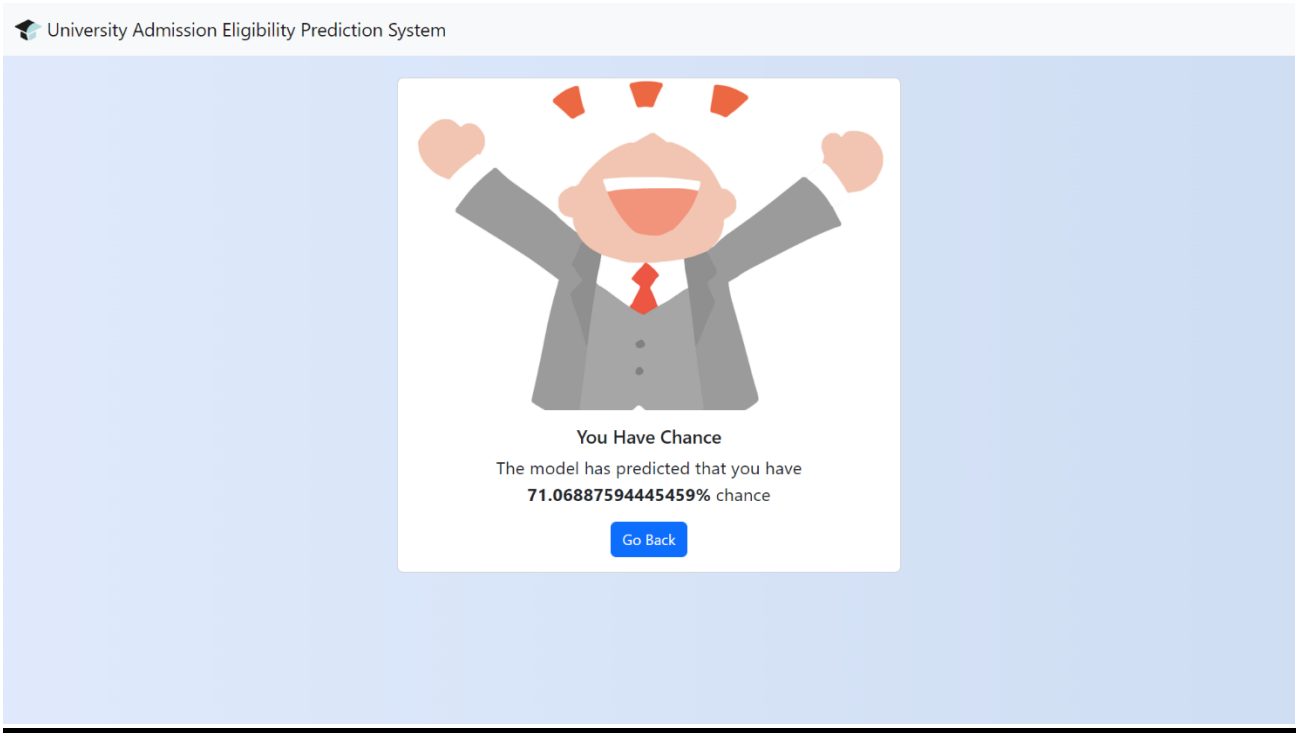



File Edit Selection View Go Run Terminal Help
Final Deliverable
EXPLORER
FINAL DELIVERABLE
dataset
output
static
css
img
js
templates
chance.html
Demo2.html
index.html
noChance.html
video
app.py
University Admission ...
university.pkl
OUTLINE
TIMELINE
Get Started
University Admission Prediction.ipynb
app.py
University Admission Prediction.ipynb > ##TITLE: UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
Code Markdown Run All Clear Outputs of All Cells Go To Restart Variables
Python 3.10.6 64-bit
lore = LogisticRegression(random_state=0, max_iter=1000)
lr = lore.fit(X_train, y_train)
[147]
Python
y_pred = lr.predict(X_test)
[148]
Python
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix
print('Accuracy Score:', accuracy_score(y_test, y_pred))
print('Recall Score:', recall_score(y_test, y_pred))
print('ROC AUC Score:', roc_auc_score(y_test, y_pred))
print('Confussion Matrix:\n', confusion_matrix(y_test, y_pred))
[150]
Python
Accuracy Score: 0.9333333333333333
Recall Score: 1.0
ROC AUC Score: 0.6


File Edit Selection View Go Run Terminal Help
Final Deliverable
EXPLORER
FINAL D...
dataset
output
static
css
img
js
templates
chance.html
Demo2.html
index.html
noChance.html
video
app.py
University Admission ...
university.pkl
OUTLINE
Get Started
University Admission Prediction.ipynb
app.py
University Admission Prediction.ipynb > ##TITLE: UNIVERSITY ADMIT ELIGIBILITY PREDICTOR
Code Markdown Run All Clear Outputs of All Cells Go To Restart Variables
Python 3.10.6 64-bit
spss-modeler_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
software_uid = client.software_specifications.get_uid_by_name('default_py3.7')
print(software_uid)
meta_props={
client.repository.ModelMetaNames.NAME: "logistic_model",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_uid,
client.repository.ModelMetaNames.TYPE: "scikit-learn_0.23"
}
[73]
Python
e4429883-c883-42b6-87a8-f419d64088cd
model_details = client.repository.store_model(model=lr, meta_props=meta_props, training_data=None)
[ ]
Python

Solutioning:





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:	300
TOFEL Score:	120
University Rating:	1
SOP:	5
LOR:	5
CGPA:	9
Research:	(•) Yes ( ) No

Predicting...

University Admission Eligibility Prediction System

You have a LOW / NO chance

The model has predicted that you only have **43.972417457648724%** chance

Go Back

CHAPTER 8

TESTING

8.1 Test Cases

Date	19-Nov-22
Team ID	PNT2022TMID34862
Project Name	Project - University Admit Eligibilit
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data
Index_page_01	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go	Score Details
Index_page_02	UI	Home Page	Verify the UI elements	Score Details	1.Enter URL and click go 2.Student the Score details	Score Details
Index_page_03	Functional	Home page	Verify user is able to log into application with Valid credentials	Score Details	1.Enter URL and click go 2.Student the Score details	worked as Details
Index_page_04	Functional	Base Page	Verify Ui Details		1.Enter URL and click go 2.Student the Score details	GIVEN Details
Index_page_05	Functional	EXpand page	Security Ui Elements		1.Enter URL and click go 2.Student the Score details	GIVEN Details
Final	Predict	Value	Verify user is able to log into application with InValid		2.Student the Score details	GIVEN Details

19-Nov-22								
PNT2022TMID34862								
Project - University Admit Eligibilit								
4 marks								
Steps To Execute	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
1.Enter URL and click go	Score Details	As the test code should display	Working as expected	Pass				Viswa B
1.Enter URL and click go 2.Student the Score details	Score Details	Application should show below UI Elements.	Working as expected	pass	Steps are not clear to follow		BUG-1234	VIGNESH K
1.Enter URL and click go 2.Student the Score details	worked as Details	User should navigate to user account homepage	Working as expected	pass	Steps are not clear to follow			MATHAN R
1.Enter URL and click go 2.Student the Score details	GIVEN Details	Application should show 'Incorrect email or password ' validation message.	Given Details	pass	The UI tools Fast			RAJAMAN S
1.Enter URL and click go 2.Student the Score details	GIVEN Details	Application should show 'Incorrect email or password ' validation message.	sCore Details	Pass	The UI tools Fast			RAJAMAN S
2.Student the Score details	GIVEN Details	Application should show 'Incorrect email or password ' validation	sCore Details	Pass	The UI tools Fast			MATHAN R

	A	B	C	D	E	F	G	H	I
1	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
2	1	337	118	4	4.5	4.5	9.65	1	0.92
3	2	324	107	4	4	4.5	8.87	1	0.76
4	3	316	104	3	3	3.5	8	1	0.72
5	4	322	110	3	3.5	2.5	8.67	1	0.8
6	5	314	103	2	2	3	8.21	0	0.65
7	6	330	115	5	4.5	3	9.34	1	0.9
8	7	321	109	3	3	4	8.2	1	0.75
9	8	308	101	2	3	4	7.9	0	0.68
10	9	302	102	1	2	1.5	8	0	0.5
11	10	323	108	3	3.5	3	8.6	0	0.45
12	11	325	106	3	3.5	4	8.4	1	0.52
13	12	327	111	4	4	4.5	9	1	0.84
14	13	328	112	4	4	4.5	9.1	1	0.78
15	14	307	109	3	4	3	8	1	0.62
16	15	311	104	3	3.5	2	8.2	1	0.61
17	16	314	105	3	3.5	2.5	8.3	0	0.54
18	17	317	107	3	4	3	8.7	0	0.66
19	18	319	106	3	4	3	8	1	0.65
20	19	318	110	3	4	3	8.8	0	0.63

8.2 User Acceptance Testing

8.2.1. Purpose of the Document

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

8.2.2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	2	1	2	10
Duplicate	0	1	2	1	4
External	4	2	2	1	9
Fixed	4	3	2	15	24
Not Reproduced	0	0	0	1	1
Skipped	0	1	0	1	2
Won't Fix	0	2	1	1	4
Totals	13	11	8	22	54

8.2.3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	32	0	6	26
Security	2	0	0	2

Outsource Shipping	4	1	0	3
Exception Reporting	8	1	1	7
Final Report Output	5	1	0	4
Version Control	2	0	0	2

CHAPTER 9

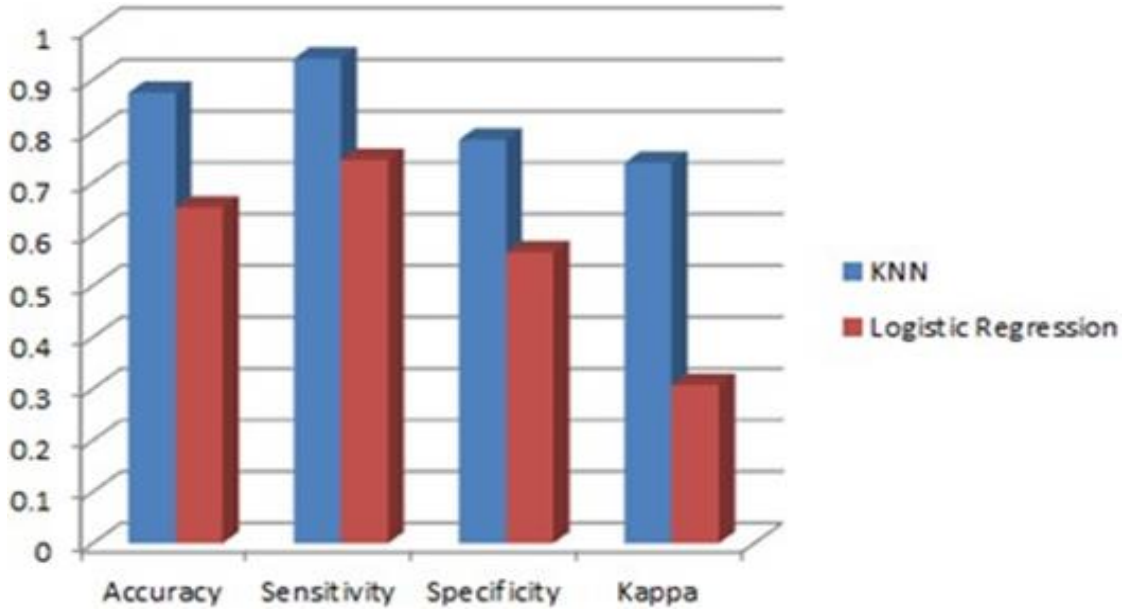
RESULTS

9.1 Performance Metrics

1. The Performance is the Accuracy of the model trained.

2.The training accuracy of the model is 92%.

3. The testing accuracy of the model is 89%.



CHAPTER 10

ADVANTAGES AND 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.

Disadvantages

- Required active internet connection.
- System will provide inaccurate results if data entered incorrectly.
- Significant investments required
- Enable to capture changes
- Privacy concerns.

CHAPTER 11
CONCLUSION

The numbers of the students seeking further higher education abroad keeps increasing every year. And this number won't go down in upcoming years either. In our technological and competitive world, the students constantly feel the need to upskill themselves and fare better than their peers and competitors every day and thus after their under graduation, a vast number of students tend to leave for higher education abroad. To facilitate an easy migration, and to provide them with a tool which can help them shortlist their potential college/university based on various parameters will not just empower them in finding their preferred institutions but also ensure that it saves their precious time as well as their hard-earned money at the same time. Many researchers have tried to come up with ideas using machine learning, data mining, and generic algorithm from time to time, each with their own benefits and drawbacks. If we go through the research done till date, the success rate of hybrid systems with the implementations of more than one algorithm concurrently is higher with low error rates, as the work done by each algorithm in any hybrid system relies on each of them only for the specific task and hence later collaboration ensures higher accuracy. In this regard, the stacked ensemble approach, which itself is based on an ensemble approach which tries to bring different algorithms together to gain better accuracy, fares better than all the other approaches discussed and reviewed in this project.

CHAPTER 12

FUTURE SCOPE

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

Details of the universities within the country can also be used for training.

APPENDIX

Source Code:

The following is the flask app code and working.

PROGRAM:

```
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 = "wf8mge_OQdwVO8ao2kmWCtfxOfLWl8442SH44V85v2Ls"
    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/8308fd4c-24a5-46ab-96fa-
263657ae4ad0/predictions?version=2022-10-18',
      json=payload_scoring,
      headers=header
    ).json()

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

    if result[0][0] > 0.5:
      return redirect(url_for('chance', percent=result[0][0]*100))
    else:
```

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

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

@app.route("/chance/<percent>")
def chance(percent):
    return render_template("chance.html", content=[percent])

@app.route("/nochance/<percent>")
def no_chance(percent):
    return render_template("noChance.html", content=[percent])

@app.route('/<path:path>')
def catch_all():
    return redirect(url_for("demo2"))

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

The following is the UI code for the application.

Chance.html

```
{% extends 'index.html' %}

{% block body %}

<div class="container text-center p-4">
    <div class="d-flex justify-content-center">

        <div class="card" style="width: 34rem;">
```

```
        <img src="..\static\img\chance.png" class="card-img-top" alt="...">
        <div class="card-body">
          <h5 class="card-title">You Have Chance</h5>
          <p class="card-text">The model has predicted that you have
<strong>{{content[0]}}%</strong> chance</p>
          <a href="/home" class="btn btn-primary">Go Back</a>
        </div>
      </div>
    </div>
</div>

{% endblock %}
```

Demo2.html

```
{% extends 'index.html' %}
{% block body %}
  <div class="p-4">
    <div class="row mb-3">
      <div class="col-4">
        <h2 class="text-responsive-h">
          Enter your details and get probability of your admission
        </h2>
        <p class="text-responsive">
          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.
        </p>
        <div class="d-flex justify-content-right">
          <img src="../static/img/animated-teach.gif" border="0" alt="..." />
        </div>
        </div>
      <div class="col-8">
        <div class="card p-2 ms-2 my-2">
          <div class="card-body">
            <h5 class="card-title pb-4">
```

```
                Enter the details
            </h5>
            <form action="/" method="post" id="theForm">
                <div class="row mb-3">
                    <label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="gre" name="gre" min="250"
max="340" placeholder="250 to 340" required>
                    </div>
                </div>
                <div class="row mb-3">
                    <label for="tofel" class="col-lg-2 col-form-label">TOFEL Score:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="tofel" name="tofel"
min="50" max="120" placeholder="50 to 120" required>
                    </div>
                </div>
                <div class="row mb-3">
                    <label for="university_rating" class="col-lg-2 col-form-label">University
Rating:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="university_rating"
step="0.01" name="university_rating" min="1" max="5" placeholder="1 to 5" required>
                    </div>
                </div>
                <div class="row mb-3">
                    <label for="sop" class="col-lg-2 col-form-label">SOP:</label>
                    <div class="col-lg-10">
                        <input type="number" class="form-control" id="sop" name="sop"
step="0.01" min="1" max="5" placeholder="1 to 5" required>
                    </div>
                </div>
```

```
                    <div class="row mb-3">
                        <label for="lor" class="col-lg-2 col-form-label">LOR:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form-control" id="lor" name="lor" step="0.01"
min="1" max="5" placeholder="1 to 5" required>
                        </div>
                    </div>
                    <div class="row mb-3">
                        <label for="cgpa" class="col-lg-2 col-form-label">CGPA:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form-control" id="cgpa" name="cgpa"
step="0.01" min="5" max="10" placeholder="5 to 10" required>
                        </div>
                    </div>
                    <fieldset class="row mb-3">
                        <legend class="col-form-label col-sm-2 pt-0">Research:</legend>
                        <div class="col-sm-10">
                            <div class="form-check">
                                <input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios1" value="1">
                                <label class="form-check-label" for="yes_no_radio">
                                Yes
                                </label>
                            </div>
                            <div class="form-check">
                                <input class="form-check-input" type="radio" name="yes_no_radio"
id="gridRadios2" value="0" checked>
                                <label class="form-check-label" for="yes_no_radio">
                                No
                                </label>
                            </div>
                        </div>
```

```
                    </fieldset>
                    <div class="row lg-3">
                        <div class="col-lg-2 mb-2 me-3">
                            <button type="submit" class="btn btn-primary"
id="button">Predict</button>
                        </div>
                        <div class="col-lg-2" id="spinner">
                            <div class="spinner-border text-primary m-1" role="status">
                                <span class="visually-hidden">Loading...</span>
                            </div>
                            <div class="spinner-grow text-primary m-1" role="status">
                                <span class="visually-hidden">Loading...</span>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
<script type="text/javascript" src="../static/js/script.js" async></script>
{% endblock %}
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no">
```

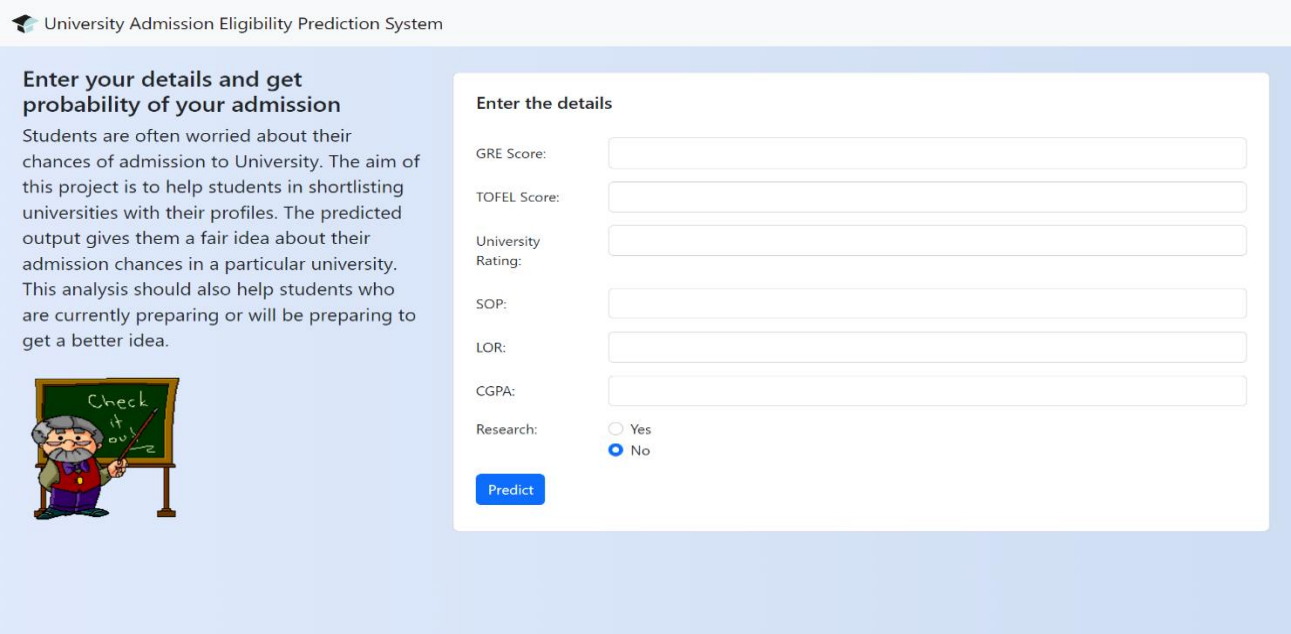
```
<link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
<title>University Admit Eligibility Predictor</title>
</head>
<body>
<nav class="navbar navbar-expand-lg bg-light">
<div class="container-fluid">
<a class="navbar-brand text-responsive-h" href="/">
<img src="..\static\img\hat.png" alt="Logo" width="30" height="24" class="d-inline-block align-
text-top ">
University Admission Eligibility Prediction System
</a>
</div>
</nav>
{% block body %}
<h1> Index Page </h1>
{% endblock %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html>
```

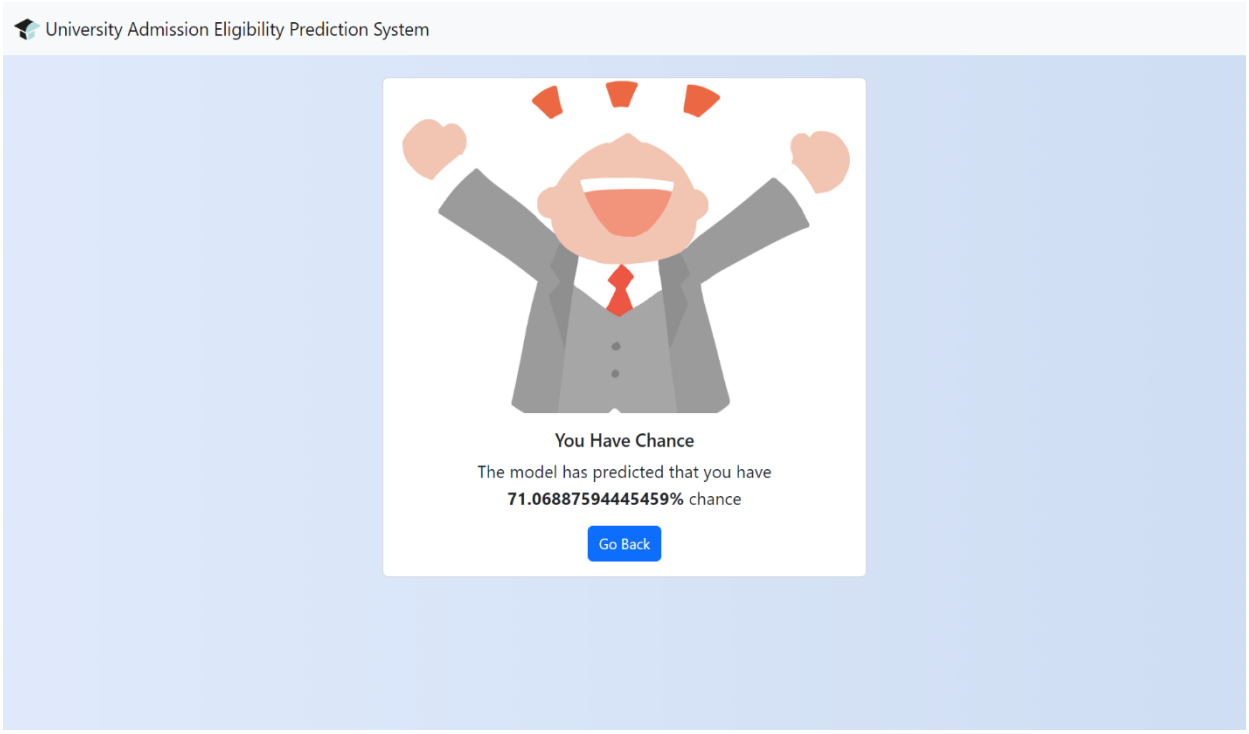
Nochance.html

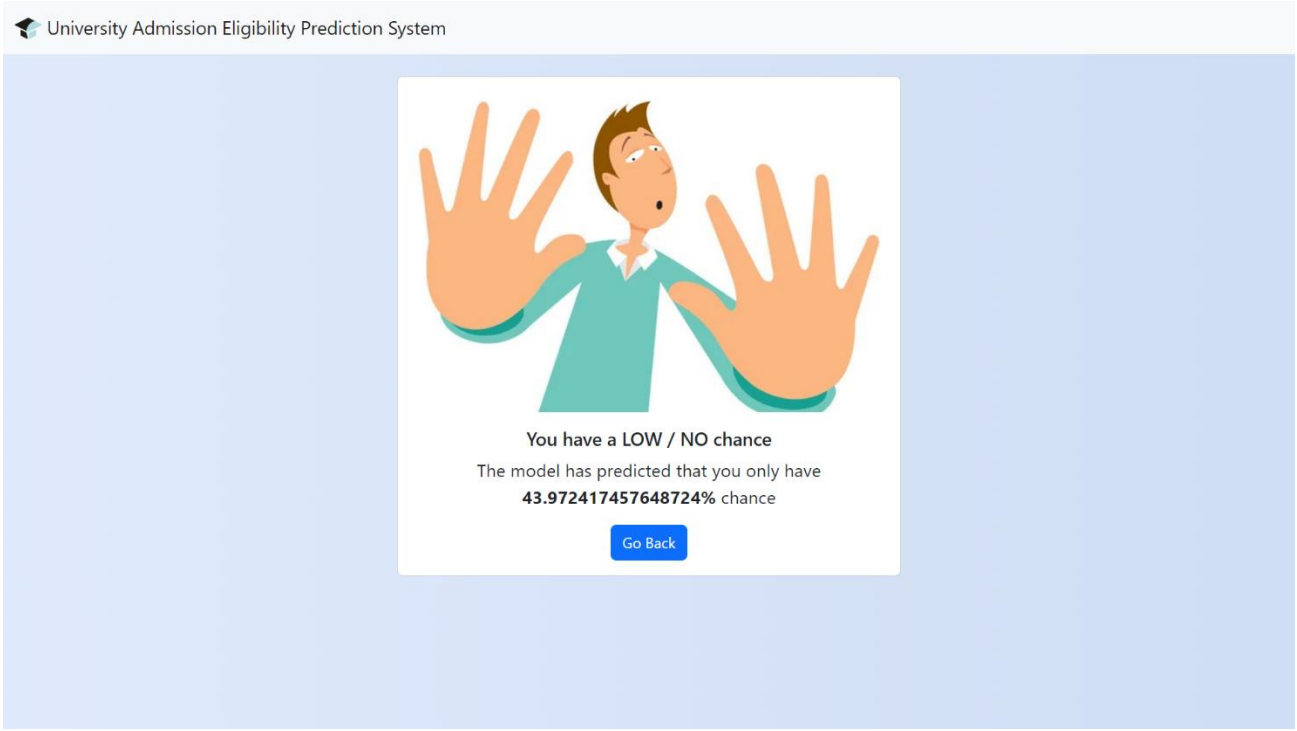
```
{% extends 'index.html' %}
{% block body %}

<div class="container text-center p-4">
```

```
    <div class="d-flex justify-content-center">
      <div class="card" style="width: 34rem;">
        <img src="..\static\img\noChance.jpg" class="card-img-top" alt="...">
        <div class="card-body">
          <h5 class="card-title">You have a LOW / NO chance</h5>
          <p class="card-text">The model has predicted that you only have
<strong>{{content[0]}}%</strong> chance</p>
          <a href="/home" class="btn btn-primary">Go Back</a>
        </div>
      </div>
    </div>
</div>

{% endblock %}
```

OUTPUT:







Github Link: https://github.com/IBM-EPBL/IBM-Project-31093-1660196119

Project Demo Link: https://drive.google.com/file/d/1hd8Zmoeof9dlaZLCpfg-rLfMsVUrWSQ-/view?usp=share_link