

PROJECT REPORT ON

UNIVERSITY ADMIT ELIGIBILITY

PREDICTOR

TEAM ID: PNT2022TMID12607

DOMAIN: APPLIED DATA SCIENCE

COLLEGE: PSG COLLEGE OF TECHNOLOGY

TEAM MEMBERS

718019Z306 - ARAVIND M

718019Z307 - BARATH KUMAR G

718019Z309 - BHOOSHAAN A

718019Z344 - SANJAI K

TABLE OF CONTENTS

1. INTRODUCTION.....	4
Project Overview	4
Purpose	4
2. LITERATURE SURVEY	5
Existing Problem	5
References	8
Problem Statement Definition	9
3. IDEATION & PROPOSED SOLUTION	10
Empathy Map Canvas	10
Ideation & Brainstorming	10
Proposed Solution	11
Problem Solution Fit	12
4. REQUIREMENT ANALYSIS	12
Functional Requirement	12
Non-Functional Requirements	13
5. PROJECT DESIGN	13
Data Flow Diagrams	14
Solution & Technical Architecture	14
User Stories	15
6. PROJECT PLANNING & SCHEDULING	16
Sprint Planning & Estimation	16
Sprint Delivery Schedule	16
Reports from JIRA	17

7. CODING & SOLUTIONING	17
Database Schema	19
8. TESTING	20
Test Cases	20
User Acceptance Testing	21
9. RESULTS	22
Performance Metrics	22
10. ADVANTAGES & DISADVANTAGES	24
11. CONCLUSION	25
12. FUTURE SCOPE	25
13. APPENDIX	26
Source Code	27
GitHub & Project Demo Link	27

1. INTRODUCTION

1.1. PROJECT OVERVIEW

The issue of student admittance must be addressed by educational institutions. This study uses machine learning techniques to forecast a student's prospects of admission to a master's degree. Knowing in advance their prospects of acceptance will benefit students. Recently graduated students sometimes lack knowledge of the requirements and processes for postgraduate admission, which may lead them to incur significant costs for consultation services to assist them determine their prospects of admission. Human computation is prone to error and prejudice. Multiple linear regression, k-nearest neighbors, random forest, and multiple layer perceptron are among the machine learning models. The Multi Layer Perceptron model performs better than other models, according to experiments.

1.2. PURPOSE

Students may choose the finest institution with the aid of the established system. The likelihood of an error in this system is smaller than it is in the current one. It is quick, reliable, and effective. prevents inconsistent and redundant data. Very user-friendly Access to data is simple. It teaches you how to further enhance your profile in order to gain admission to the institution of your choice. It could help you decide whether to take the GRE again if you want to increase your chances of being accepted to the university of your choice. It is challenging for students from rural backgrounds to finish the required research and construct a preference list. They will gain from this idea.

2. LITERATURE SURVEY

When it comes to international students the first choice of the majority of them is the United States of America. With the majority of the world's highly reputed universities, a wide range of courses offered in every sector, highly accredited education system and teaching, scholarships provided to students, best job market and many more advantages make it the dream destination for the international 1 students. According to research, there are above 8 Million international students studying in more than 1700 public and 2500 private universities and colleges across the USA. (MasterPortal (2017)).

2.1. EXISTING PROBLEM

The main limitation of earlier research in this field is that it did not take into account all the variables that will affect a student's admission, such as TOEFL/IELTS, SOP, LOR, and undergrad score. Earlier research in this field used the Naive Bayes algorithm to evaluate the success probability of a student application into a specific university. The application submitted by the university's international students was evaluated using a decision support network built using the Bayesian Networks Algorithm. This model was created to predict how prospective students would perform by comparing their performance to that of university students who are currently enrolled. In this manner, the model made predictions about whether the prospective student should be admitted to the university based on a variety of student test scores.

Hybrid Recommender System for Predicting College Admission:

This paper proposes a hybrid recommender system for predicting college admissions based on various factors such as historical college admissions data, students' academic merits, background, student records, and college criteria. It provides insight into a hybrid model based on knowledge discovery rules and data mining techniques to predict the admission chances of any given student. The system proposed to hold a web portal for aspiring students to register themselves, which in turn assigns a unique ID number and password to each student. The students are required to enter their personal details, academic history, and desired programs for enrolling in the web portal. These details are used as input to the trained model to make predictions about the student's admission.

The system also embeds a sorter and a filter, combinedly known as the Track Recommender, which sorts the students to several university study tracks available and rearranges them based on outstanding backlogs. The HRSPCA

uses knowledge discovery rules to cluster students and recommends colleges under reach for each student.

CAPSLG:

In the paper “College Admission Predictor and Smart List Generator”, the authors have discussed an application that helps students to make correct preferences of colleges. This application has two main components. The college eligibility predictor and a smart list generator.

- The college eligibility predictor takes in input as the history of cut-off records of the colleges and trains on this data to determine which college the student is likely to get admission to. Not just the cut-off data, the system also examines a student's academic achievements, history, and requirements for college admission. Based on it, it forecasts the likelihood that a student will enroll in a university college.
- Further, the student might create the list of colleges that might be required to be filled out as choices throughout the admissions process using the smart list generator.
- Additionally, the system would receive user feedback, which would be beneficial for evaluating predictions and enhancing the performance factor.

On the whole, The program predicts the best-suited college based on the users' preferences, courses, areas, and cut-off of colleges once they enter their academic scores. The application also provides a list of institutions based on the aforementioned factors, which will assist students in narrowing down their options and more accurately completing their admissions applications.

Primarily, in the paper, the authors have performed an analytical study of various ML Algorithms such as Random Forest, AdaBoost, and Decision Tree by checking their classification accuracies on Breast Cancer, Iris, and Wine Datasets. They concluded that the Adaboost model performed the best and thus, the Ensemble AdaBoost Classifier from the scikit-learn library of Python is used for the classification of the data. The AdaBoost algorithm divides the input into two classes(binary classifier). Based on the classification from numerous decision trees, it categorizes the incoming data. The algorithm's classification makes up the majority of the classification produced from the chosen number of decision trees. As a result, the AdaBoost algorithm is more accurate. Additionally, it extracts the features that are more important for predicting the outcomes. The model is actually pre-trained and pickled. The model is then loaded from the stored pickle and used for prediction tasks, which improves efficiency in terms of

time.

The Python-based Django web page rendering framework serves as the foundation for the web application. The Django framework enables the user interface and Python variables to be integrated. HTML and CSS were used to create and style the user interface, which controlled the flow of the application based on user interaction.

College Admission Prediction using Ensemble Machine Learning Models:

This paper proposes an Education based Prediction System that allows the students to decide on which colleges to apply to with their scores being fed into the application. The dataset consists of the University name, GRE score, TOEFL score, AWA scores, letters of recommendation, statement of purpose, and CGPA scores. It aims at saving time by applying to appropriate colleges based on the predicted admission feasibility. They proposed a novel method by utilizing machine learning models such as Neural Networks, Linear Regression, Decision Trees, and Random Forest. The algorithms are experimented and the key performance indicators are compared in order to choose the best mode. The application takes in the scores of a particular student and the university as input and generates a probability score as to how likely the student is to get into the university based on the scores.

Linear regression:

Regression models are employed to describe a relationship between many variables by fitting observed data into a straight line. It is used to determine the connection between two numerical variables. Regression models differ in terms of the type of relationship between independent and dependent variables, the number of variables employed, and the variables considered. A linear function that predicts continuous values as a function of independent variables, admission intake, and preferences of students.

Neural Network:

Neural networks try to recognize patterns in the underlying data through a process similar to that employed by the human brain. It consists of neurons organized in a series of layers which together constitute the entire artificial neural network. The number of layers, the number of neurons in each layer, and the

activation function to be used are experimentally determined so as to get the best possible result. The input layer is fed in with the scores of the individual student along with the required university identifier. The data is processed in a series of layers and the output layer uses a softmax layer to restrict the probability of admission in the range of 0 to 1.

Decision Trees:

The decision tree algorithm can be used for solving classification and regression problems. It splits the dataset based on the output class label and attributes' values for the records. It assigns class labels for each leaf node and internal nodes contain test conditions on particular attributes.

Random Forest:

Decision trees are built on multiple different samples and the majority votes for average. It works successfully for classification problems due to the fact that any huge quantity of moderately uncorrelated trees working as a body will outperform any of the individual constituent models.

It was found that linear regression outperformed other models and hence has the highest accuracy. The decision tree had the least accuracy of all the models and is spread with a lot of outliers, thus the model was inaccurate.

2.2. REFERENCES

- [1] Ragab, A.H.M., Hybrid recommender system for predicting college admission, Intelligent Systems Design and Applications (ISDA), 29 Nov. 2012, 107-113.
- [2] CAPSLG: College Admission Predictor and Smart List Generator By Kiran Kumari, Meet Kataria, Viral Limbani, Rahul Soni
- [3] College Admission Prediction using Ensemble Machine Learning Models By Vandit Manish Jain, Rihaan Satia

2.3. PROBLEM STATEMENT DEFINITION

In today's world, it can be seen that many students pursue their higher education in foreign countries away from their native countries. This being the case, students may not be aware of the admission procedures, criteria,

requirements, and information regarding admission into various universities. In order to know all this information, students seek help from consultancy firms to help them get admission to universities that best suit their profiles. But, this requires investment from students for consultancy fees. Thus, this project aims at developing an application that uses machine learning-based algorithms to determine the feasibility of a particular student's profile being eligible for university admission. The main objective is to save the time and money spent by the students at education consultancy firms. Moreover, if the students apply only to those universities where he/she has a genuine chance of admission would reduce the application process.



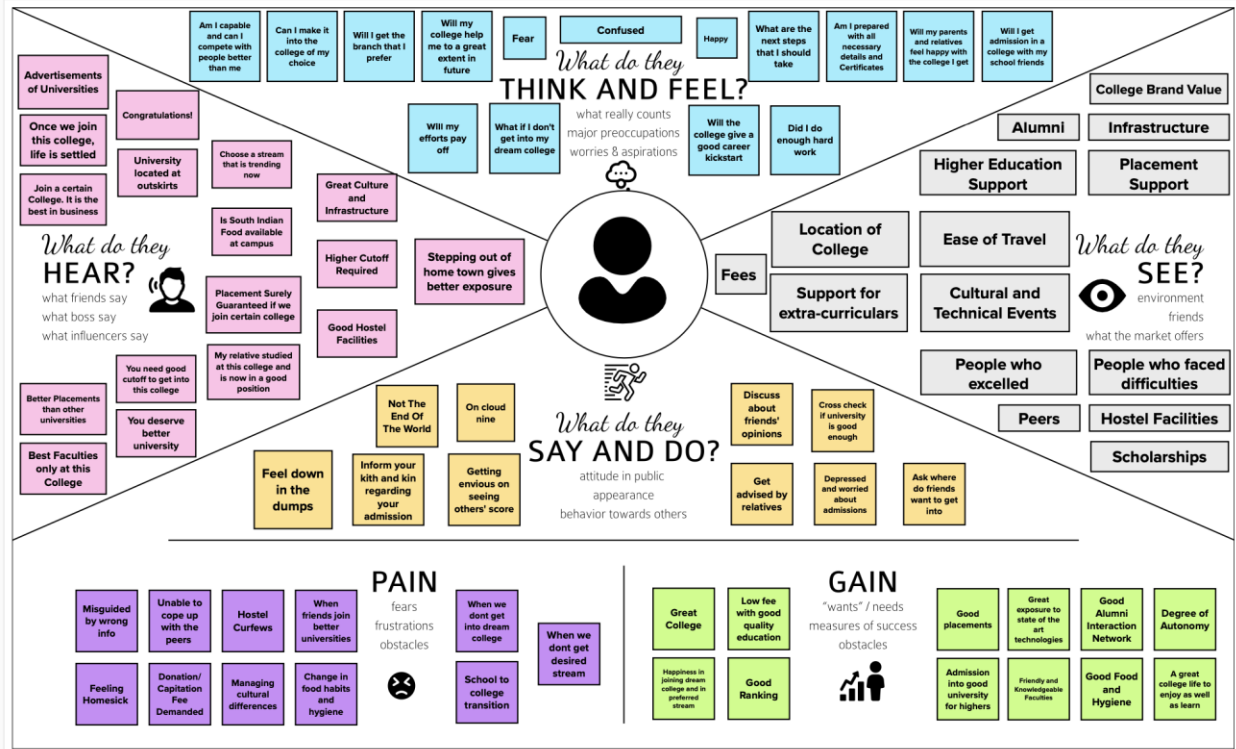
3. IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS

University Admit Eligibility Prediction

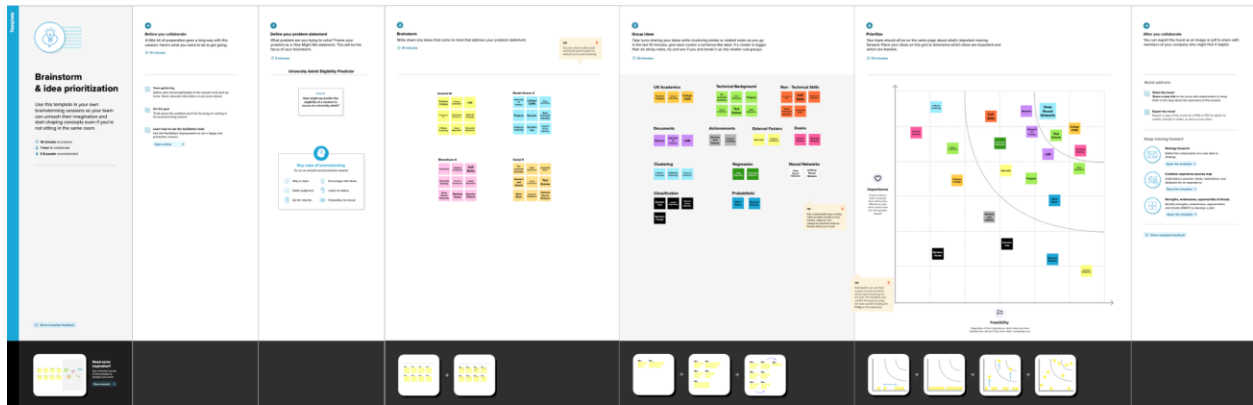
Understanding Issues Faced by Students during their Process of Applying to Universities

Empathising with students by focusing them and taking a view by putting us in their shoes



3.2. IDEATION AND BRAINSTORMING

This task of shortlisting the universities where the student has high chances of admission is difficult for mainly 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 these students have to pay a huge amount of fee to the education consultant.

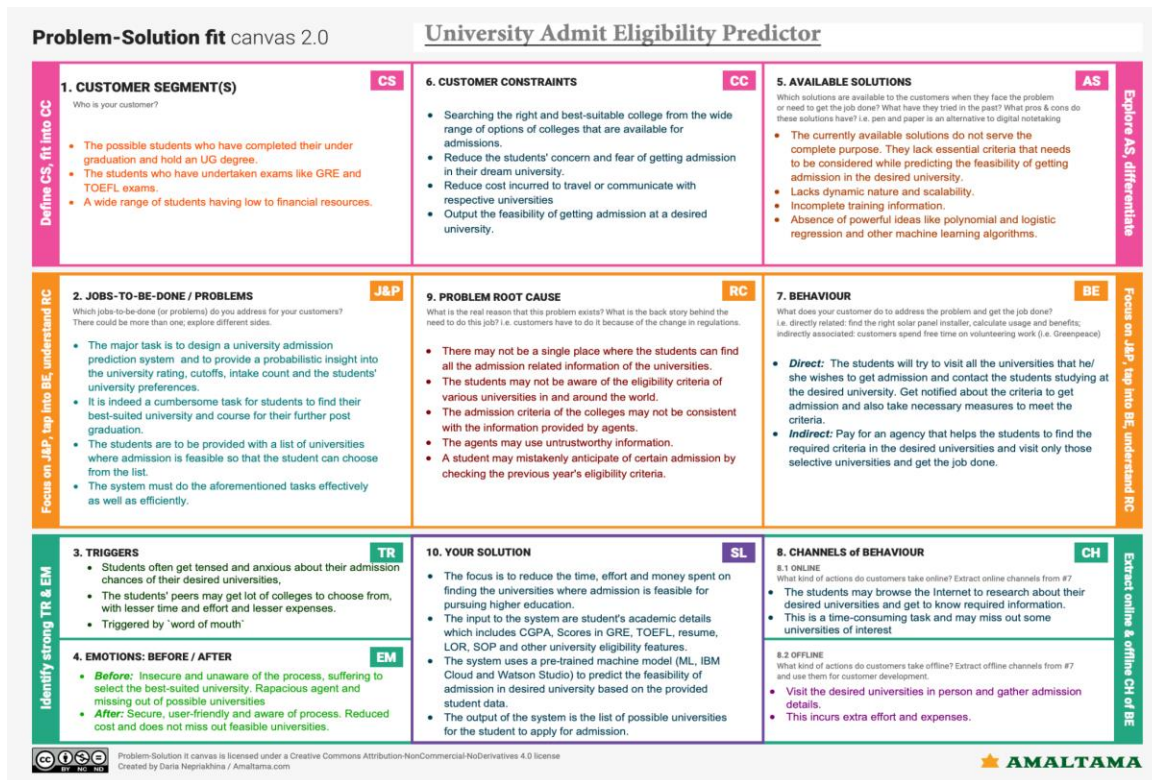


Link: tinyurl.com/37sxhwnz

3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	This project aims at developing an application that uses machine learning-based algorithms to determine the feasibility of a particular student's profile being eligible for university admission. The main objective is to save the time and money spent by the students at education consultancy firms. Moreover, if the students apply only to those universities where he/she has a genuine chance of admission would reduce the application process.
2.	Idea / Solution description	Our project will assist UG graduates in getting into shortlisted colleges for master's programmes based on their GRE, CGPA, and TOEFL scores. If the expected production gives them a good picture of their prospects of admission to the university. This study will also assist students who are presently preparing to have a better understanding. It will also provide students with information on the university's research prospects, admissions procedure, courses offered, and noteworthy alumni.
3.	Novelty / Uniqueness	The project website can identify numerous amenities available at universities and provide directions to the university where it is located. You can also apply for scholarships and financial aid. By using Machine learning models like Regression models, the probability of a student getting admission at a desired university is predicted.
4.	Social Impact / Customer Satisfaction	This solution will ease their stress about being admitted to their preferred university, as well as minimize student anxiety. And this solution will deliver better outcomes for students who are deciding whether or not to attend university.
5.	Business Model (Revenue Model)	In addition, revenue can be generated by advertising the GRE / TOEFL coaching centres. And the University shall fund the website in order to maintain and progress it. The universities can also find a way to advertise in the website in order to increase the admissions.
6.	Scalability of the Solution	The solution proposed will be deployed as web-application. So, it is easily accessible by anyone who has internet services and has no specific software and hardware specifications. The dataset used for model training can be scaled according to the available universities' admission data.

3.4. PROPOSED SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

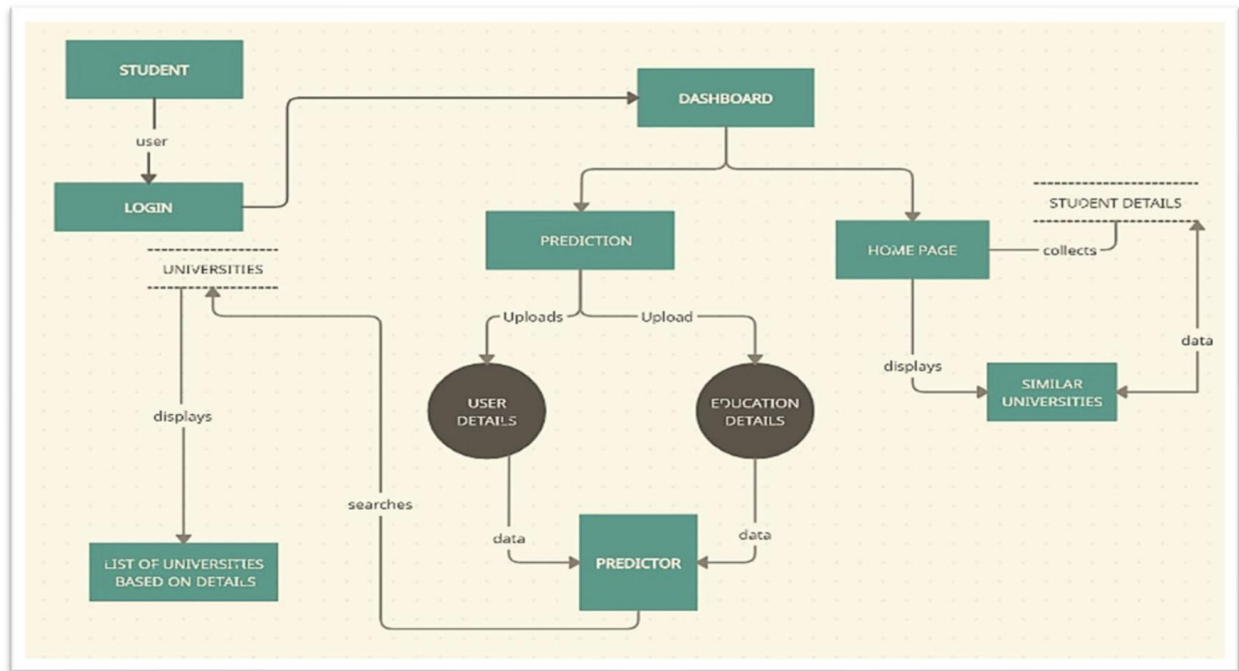
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Users can register by filling details form. Users can register through Gmail.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Details	Upload the required documents for admission prediction <ul style="list-style-type: none"> ❖ Curriculum Vitae (CV) or Resume ❖ Letter of Recommendation (LOR) ❖ GRE and TOEFL Score Marksheet ❖ Statement of Purpose (SOP) ❖ Research Publications (DOI)
FR-4	User Requirements	In order to begin with process of predicting the admit eligibility of a candidate at a university, the activities to be done are: <ul style="list-style-type: none"> ❖ Submit all the relevant documents in the specified location at the website. ❖ Based on the submitted information, the UAEP system collects all the information necessary for prediction. ❖ The list of universities that are available in the website are displayed to the user.

4.2. NON-FUNCTIONAL REQUIREMENTS

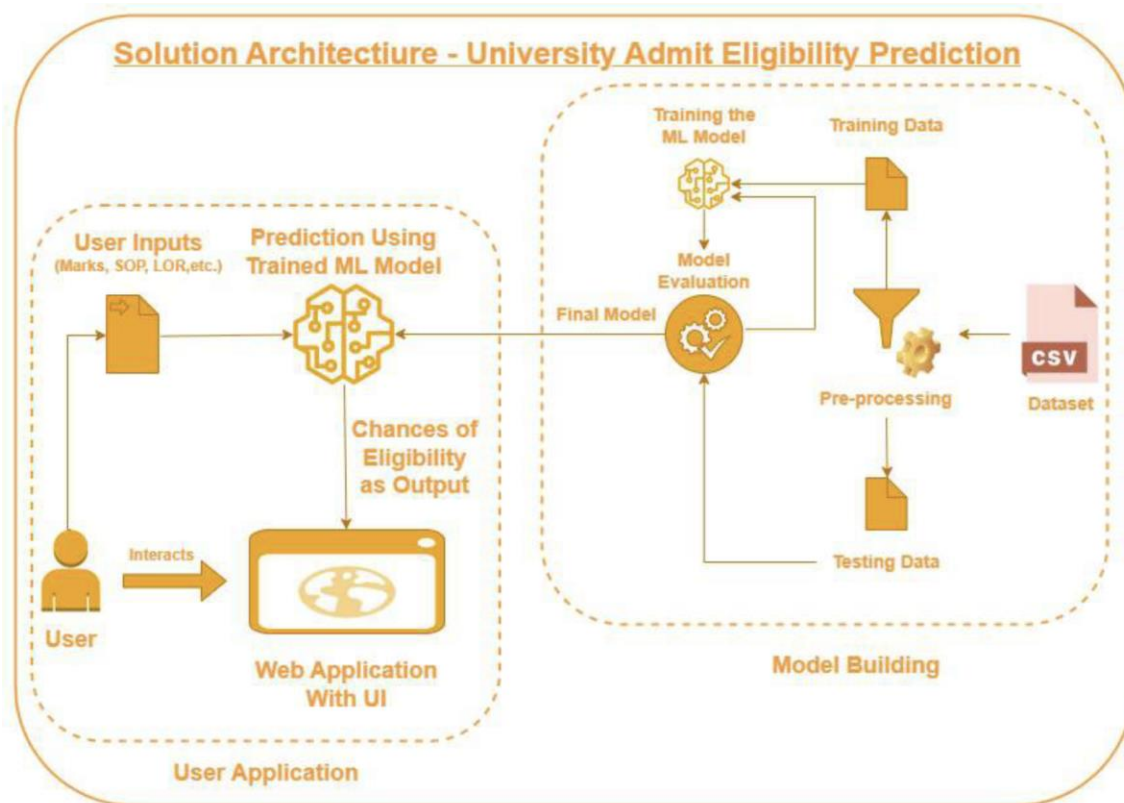
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none"> ❖ There should not be any technical dependency required for the system. In other words, even a naïve user should be able access the system with ease. ❖ The user-interface of the website should be designed in such a way each page focuses on recognize over recall. ❖ The system should be user-friendly and should display hints wherever necessary. ❖ All the input fields present in the website should be self-explanatory. ❖ The time taken for the webpage to load and display content should be less than 15 seconds. ❖ Reduced focus on short term memory load. ❖ The website must be responsive and compatible across all the types of devices in which the site is accessed.
NFR-2	Security	<ul style="list-style-type: none"> ❖ The backend database must be periodically backed up to ensure the consistency and reliability. ❖ The system should be able to rollback to normal state on occurrence of any errors or faults. ❖ The system should provide authentication and authorization of the users and allow only authorized ones to utilize the services of the site.
NFR-3	Reliability	<ul style="list-style-type: none"> ❖ The system should be able to function for 24 x 7 hours for a week. ❖ The importance of data being stored and costs involved in damages force the system to be more reliable. ❖ The system must be able to quickly handle and recover from any of the failures or crashes.
NFR-4	Performance	<ul style="list-style-type: none"> ❖ The website must efficiently handle by servicing the request as soon as possible. ❖ The internet requirements of the user should not drastically affect the system performance. ❖ The search and filter operations can be made quick by the usage of indexes in the database.
NFR-5	Availability	<ul style="list-style-type: none"> ❖ Data redundancy should be minimized. ❖ The system should be less error-prone. ❖ Fast and efficient usage. ❖ The system should be able to function for 24 x 7 hours for a week.
NFR-6	Scalability	<ul style="list-style-type: none"> ❖ The system should be able to handle a reasonable amount of user-traffic and provide results efficiently. ❖ The admission or intake season is the crucial time where the system tends to experience huge amount of traffic ❖ It must be able to manage a number of concurrent users.

5. PROJECT DESIGN

5.1. DATA FLOW DIAGRAM



5.2. SOLUTION & TECHNICAL ARCHITECTURE



5.3. USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	Users can register for the application by entering my email, password, and confirming my password.	Access user's account / dashboard	High	Sprint-1
		USN-2	Users will receive confirmation email once I have registered for the application	Receive confirmation email & click confirm	High	Sprint-1
		USN-3	Users can register for the application through Facebook	Register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	Users can register for the application through Gmail	Register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	Users can log into the application by entering email & password	Access the dashboard	High	Sprint-1
	Dashboard	USN-6	Users can enter the scores.	Predict eligibility	High	Sprint-2
Customer (Web user)	Registration	USN-7	Users can register for the application by Entering his/her email, password, and confirming the password.	Access user's account / dashboard	High	Sprint-1
		USN-8	Users will receive confirmation email once I have registered for the application	Receive confirmation email & click confirm	High	Sprint-1
		USN-9	Users can register for the application through Facebook	Register & access the dashboard with Facebook Login	Low	Sprint-2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-10	User can register for the application through Gmail	Register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-11	Users can log into the application by entering email & password	Access the dashboard	High	Sprint-1
	Dashboard	USN-12	Users can enter the scores.	Predict eligibility	High	Sprint-2
Customer Care Executive	Support	USN-13	As a Customer Care Executive, responding to queries via telephone, live chat etc.	Send Immediate response	Medium	Sprint-3
		USN-14	As a Customer Care Executive, Ask for and act on customer feedback	Thank users their valuable feedback	High	Sprint-2
		USN-15	As a Customer Care Executive, analyze customer data and communication to adjust customer care strategies.	Look into that issue soon and try to rectify it	Low	Sprint-3
Administrator	Administrative functions	USN-16	As an Administrator, design, develop, maintain and troubleshoot websites.	No issues found	High	Sprint-3
		USN-17	As an Administrator, view and manage user permissions in an application.	Allows the user to manage permissions	Low	Sprint-4
		USN-18	As an Administrator, implementing user protocols & creating backups.	Data is synced & later recovered	Medium	Sprint-4
		USN-19	As an Administrator, resolving software problems & updating new features.	Update to new features	High	Sprint-4

6. PROJECT PLANNING AND SCHEDULE

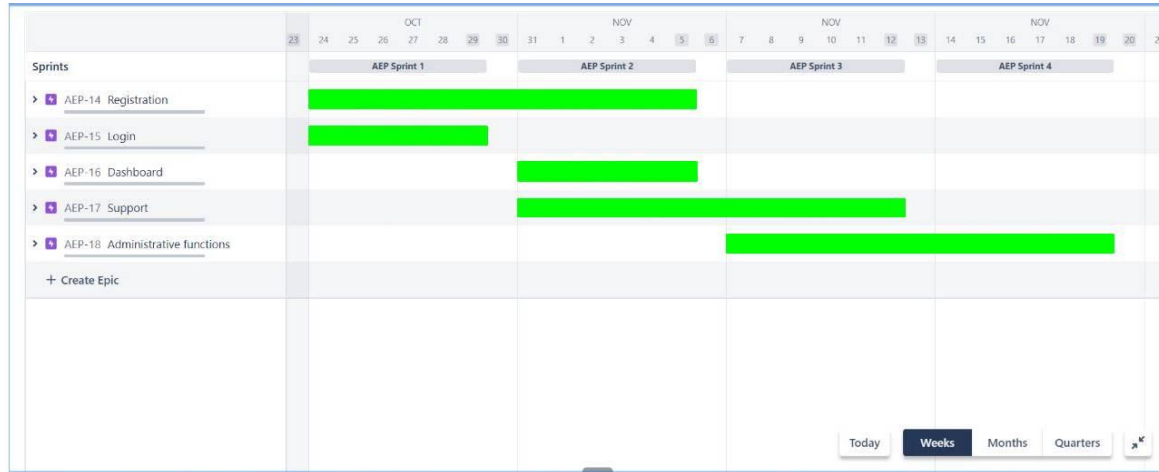
6.1. SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I will be able to register my application by entering my email, password, and confirming my password.	2	High	Aravind M
Sprint-1		USN-2	As a user, I will be able to receive an email confirmation after registration.	1	High	Barath Kumar G
Sprint-2		USN-3	As a user, I can register for the application through Gmail.	2	Low	Sanjai K
Sprint-1		USN-4	As a user, I can register for the application by entering details by self.	2	Medium	Bhooshaan A
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Bhooshaan A
	Dashboard					

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	4 Days	1 Nov 2022	4 Nov 2022	20	1 Nov 2022
Sprint-2	20	3 Days	5 Nov 2022	6 Nov 2022	20	5 Nov 2022
Sprint-3	20	3 Days	7 Nov 2022	9 Nov 2022	20	7 Nov 2022
Sprint-4	20	5 Days	10 Nov 2022	15 Nov 2022	20	10 Nov 2022

6.1. REPORTS FROM JIRA



7. CODING AND SOLUTIONING

```
from flask import Flask , request , redirect
import os
import pickle
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import metrics
from flask import render_template
import requests

API_KEY = "3-djqxBIyr1X0buXEcoEXxf00-a4C3qPf9L53Soe4Vv"
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__)

@app.route("/")
def index():
    print(os.getcwd())
    return render_template("index.html",score=100)

@app.route("/predict",methods=['POST'])
def predict():
    gre_score=request.form["gre"]
```

```

toefl_score=request.form["toefl"]
ur_value=request.form["ur"]
lor_score=request.form["lor"]
sop_score=request.form["sop"]
cgpa_value=request.form["cgpa"]
rp_value=request.form["rp"]
t=[[int(gre_score),int(toefl_score),int(ur_value),float(sop_score),float(lor_score),float(cgpa_value),int(rp_value)]]
payload_scoring = {"input_data": [{"fields": [['GRE Score','TOEFL Score','University Rating','SOP','LOR','CGPA','Research']], "values": t
}}}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/e327ebc2-eb51-46a5-b2c3-57c81badc74e/predictions?version=2022-11-15', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})

output= response_scoring.json()["predictions"][0]["values"][0][0]
op=(output*100)

while op>=100:
    op-=2
while op<5:
    op+=5
op=round(op,2)
if op>=50:
    return render_template("success.html",score=op)
else:
    return render_template("fail.html",score=op)

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

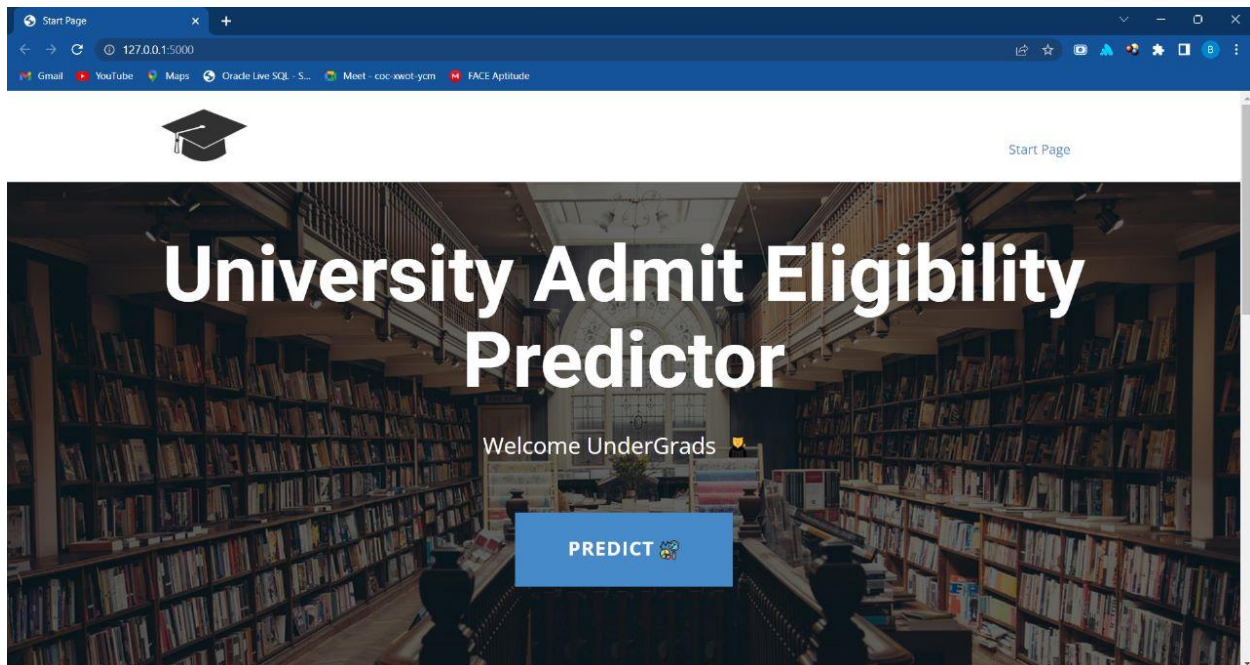
```

DATASET SCHEMA:

The dataset consists of the academic attributes of the students who have opted for admissions in various universities. The attributes are:

- GRE Score
- TOEFL Score
- University Rating
- Statement of Purpose (SOP)
- Letter of Recommendation (LOR)
- CGPA
- Research Papers published
- Admit Probability

SCREENSHOTS



Prediction Input Data

GRE Score

TOEFL Score

University Rating

Letter of Recommendation

Statement of Purpose

CGPA

Research Papers

Predict

* All the fields are required for accurate eligibility prediction

8. TESTING

8.1. Test Cases

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
11	325	106	3	3.5	4	8.4	1	0.52
12	327	111	4	4	4.5	9	1	0.84
13	328	112	4	4	4.5	9.1	1	0.78
14	307	109	3	4	3	8	1	0.62
15	311	104	3	3.5	2	8.2	1	0.61
16	314	105	3	3.5	2.5	8.3	0	0.54
17	317	107	3	4	3	8.7	0	0.66
18	319	106	3	4	3	8	1	0.65

8.2. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is a sort of testing conducted by the end user or customer to verify/accept the software system before it is moved to the production environment. UAT is performed at the end of the testing process, following functional, integration, and system testing. This product's user acceptance has not been sufficiently surveyed to provide a meaningful judgement. This product's theoretical and hypothetical acceptance is assessed to be high enough to determine that it is usable and valuable.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	2	1	3
Duplicate	1	0	0	0	1
External	0	0	1	0	1
Fixed	0	1	1	0	2
Not Reproduced	0	1	0	0	1
Skipped	0	0	0	0	0
Won't Fix	0	1	0	0	1
Totals	1	3	4	1	9

Selecting from Drop down	5	0	0	5
Final Report Output	30	0	10	20
Version Control	5	0	2	3

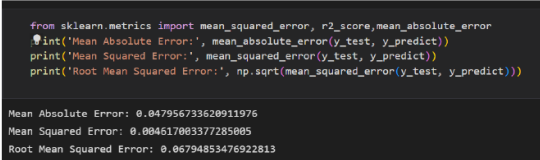
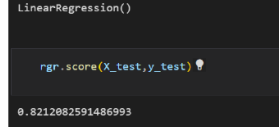
Section	Total Cases	Not Tested	Fail	Pass
View Home Page	10	0	3	7
Enter the scores	20	0	3	17
Click Submit button	2	0	0	2
Image displayed	10	0	4	6

9. RESULTS


9.1. PERFORMANCE METRICS

```
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
print('Mean Absolute Error:', mean_absolute_error(y_test, y_predict))
print('Mean Squared Error:', mean_squared_error(y_test, y_predict))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_predict)))
```

Mean Absolute Error: 0.047956733620911976
Mean Squared Error: 0.004617003377285005
Root Mean Squared Error: 0.06794853476922813

S.No.	Parameter	Values	Screenshot
1.	Metrics	Regression Model: MAE - 0.047956733620911976, MSE - 0.004617003377285005, RMSE - 0.06794853476922813	
2.	Tune the Model	Hyperparameter Tuning – Cross Validation and Epochs	

PREDICTING PERCENTAGE



Prediction Input Data

GRE Score

300

TOEFL Score

100

University Rating

2

Letter of Recommendation

5

Statement of Purpose

4.5

CGPA

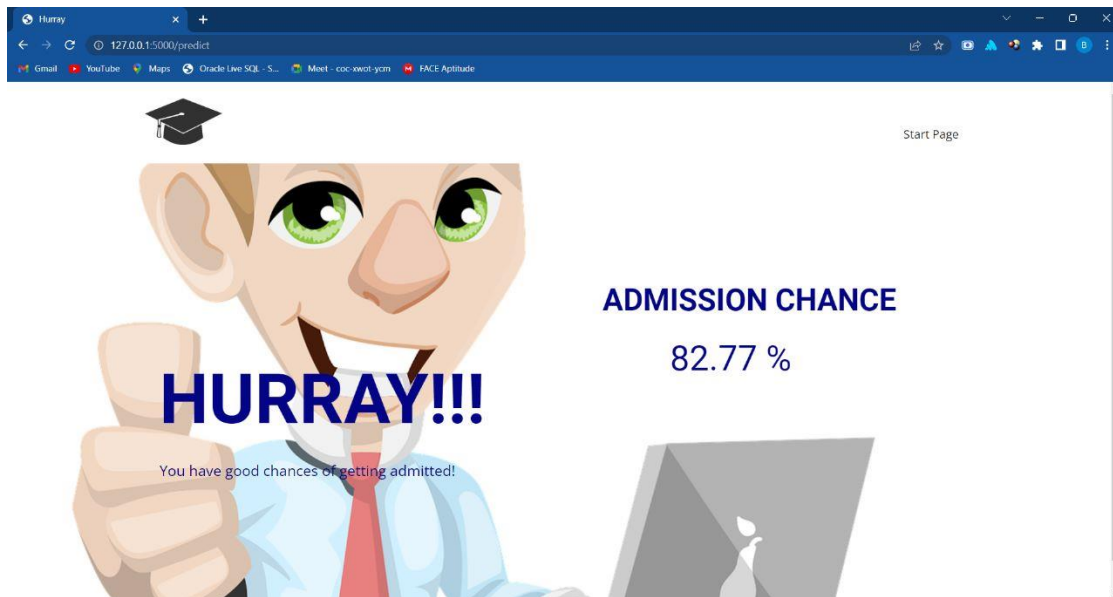
9.7

Research Papers

1

PREDICT

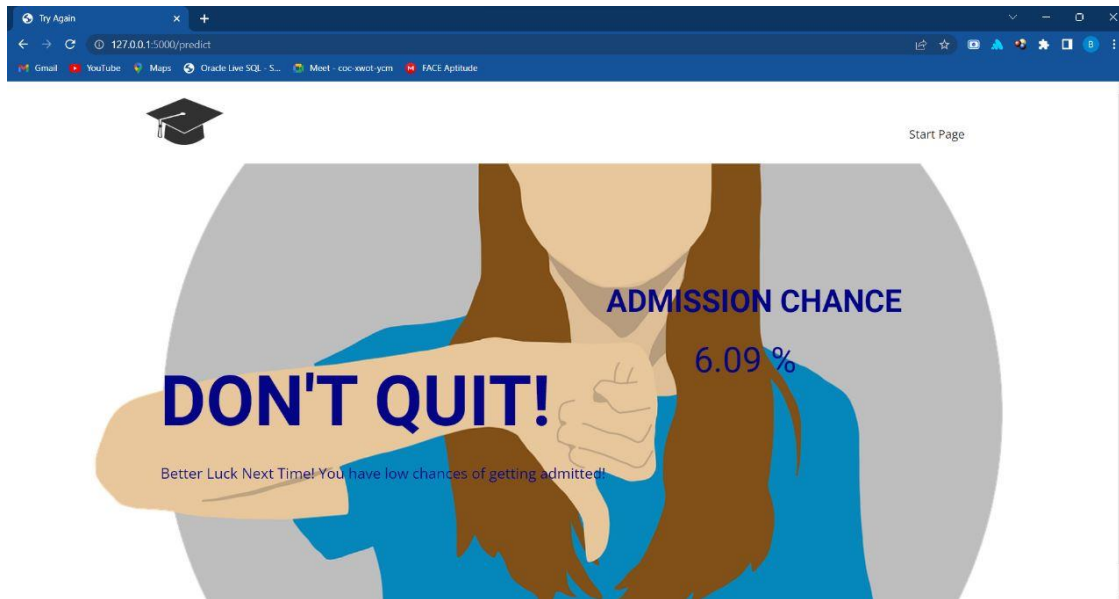
* All the fields are required for accurate eligibility prediction



A screenshot of a web browser window showing a 'Prediction Input Data' form. The browser's address bar shows '127.0.0.1:5000'. The form is titled 'Prediction Input Data' and includes a 3D character pointing at a chalkboard on the left. The form fields are as follows:

GRE Score	100
TOEFL Score	70
University Rating	1
Letter of Recommendation	2
Statement of Purpose	1
CGPA	7.2
Research Papers	0

Below the form is a blue button labeled 'PREDICT'. At the bottom, a note states: '* All the fields are required for accurate eligibility prediction'.



10. ADVANTAGES

- It helps student for making decision for choosing a right college.
- Here the chance of occurrence of error is less when compared with existing system.
- It is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data.
- It would be the easiest mode to predict the university/colleges person is applicable for as well as it would be unbiased and totally transparent.
- Additionally living expense of the area where colleges/university is located would also be provided on website.

DISADVANTAGES

- Required active internet connection.
- System will provide inaccurate results if data entered incorrectly.
- Other factors such as changes in policies by the university or by the country can also affect chances of admissions in a way that is beyond the scope of this project.
- Admissions also depend on the individual university's policy

regarding the intake of foreign students and is not modeled by our system.

11. CONCLUSION

Student admission problem is very important in educational institutions. In this project addresses machine learning models to predict the chance of a student to be admitted. This will assist students to know in advance if they have a chance to get accepted. In this paper, machine learning models were performed to predict the opportunity of a student to get admitted to a master's program. The machine learning models included are multiple linear regression, k nearest neighbor, random forest, and Multi-layer Perceptron. Experiments show that the Multi-layer Perceptron model surpasses other models. As for the future work, more models can be conducted on more datasets to learn the model that gives the best performance.

12. FUTURE SCOPE

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

- This can be implemented in less time for proper admission process.
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system.
- The scope of this project is a web application that allows users to enter their academic data and get predictions of their chances of admissions in the university tier of their choosing.
- It also provides an analysis based on the data set used that shows how the different affect chances of admissions.
- A Database will also be implemented for the system so that students can save their data and review and edit it as they progress with the most recent predictions being saved with their profile.
- Future work in the project could include weighing in the features that have been ignored as of yet like percentage seats for Foreign Students.
- Other criterion's like Co-curricular achievements, Leadership positions held, job experience, etc can also be included as metrics for the model.

13. APPENDIX

13.1. SOURCE CODE

```
14. from flask import Flask , request , redirect
15. import os
16. import pickle
17. from sklearn import metrics
18.
19. from flask import render_template
20. import requests
21.
22. API_KEY = "3-djqxBIyr1X0buXEcoEXxxf00-a4C3qPf9L53Soe4Vv"
23. token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
24. data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-
    type:apikey'})
25. mltoken = token_response.json()["access_token"]
26.
27. header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
    mltoken}
28.
29. app=Flask(__name__)
30.
31. @app.route("/")
32. def index():
33.     print(os.getcwd())
34.     return render_template("index.html",score=100)
35.
36. @app.route("/predict",methods=['POST'])
37. def predict():
38.     gre_score=request.form["gre"]
39.     toefl_score=request.form["toefl"]
40.     ur_value=request.form["ur"]
41.     lor_score=request.form["lor"]
42.     sop_score=request.form["sop"]
43.     cgpa_value=request.form["cgpa"]
44.     rp_value=request.form["rp"]
45.     t=[[int(gre_score),int(toefl_score),int(ur_value),float(sop_score),flo
    at(lor_score),float(cgpa_value),int(rp_value)]]
46.     payload_scoring = {"input_data": [{"fields": [['GRE Score','TOEFL
    Score','University Rating','SOP','LOR','CGPA','Research']], "values": t
    }}
47.     response_scoring = requests.post('https://us-
    south.ml.cloud.ibm.com/ml/v4/deployments/e327ebc2-eb51-46a5-b2c3-
    57c81badc74e/predictions?version=2022-11-15', json=payload_scoring,
48.     headers={'Authorization': 'Bearer ' + mltoken})
49.
```

```
50.     output= response_scoring.json()["predictions"][0]["values"][0][0]
51.     op=(output*100)
52.     while op>=100:
53.         op-=2
54.     while op<5:
55.         op+=5
56.     op=round(op,2)
57.     if op>=50:
58.         return render_template("success.html",score=op)
59.     else:
60.         return render_template("fail.html",score=op)
61.
62. if __name__ == "__main__":
63.     app.run(debug=True)
```

13.2. GitHub Repo and Demo Video Link

Git: <https://github.com/IBM-EPBL/IBM-Project-13892-1659534676>

Demo: <https://www.youtube.com/watch?v=N-ah3yZ5fws>